NGG

Notes Graphics user interface Gallery

Leong Pek Ling

WET 97019

Session 1999/2000

Supervisor

Dr. Siti Salwah Salim

This thesis is submitted to

Faculty of Computer Science and Information Technology

University Of Malaya

In partial fulfillment of the requirement for the

Bachelor of Information Technology

ABSTRACT

Nowadays, the user interfaces for most system involves graphics. Notes Graphics User Interface Gallery (NGG) is a toolkit developed to support interactive user-interface design for the applications of Lotus Notes. This toolkit is used in Lotus Notes application to assist the programmers to design interfaces or menus for their application.

The development of NGG proves that it is more attractive to apply graphics on user interface. Before developing NGG, researches about all the features for user interface in Lotus Notes was done.

Notes GUI Gallery consists of two components, where the first component is a library of graphics, such as lines, buttons, icons and pictures. These graphics can be copied and pasted into Lotus Notes design elements such as Navigators, Layout Regions and Forms. Another component of NGG is a collection of templates for navigators and forms. The objective of this project is to assist developers to build applications in a faster and easier way so that they can focus on the system design and development tasks.

NGG was developed using Lotus Notes Release 4.6 software to create and store database of templates and graphics. The graphics section of this toolkit can be used for other applications rather than Lotus Notes. It is believed that toolkit of this nature will gradually become essential to everyone in the future.

Acknowledgement

Many individuals have assisted and helped me in preparing myself for the final year project. I would like to express my deepest gratitude and thanks to my project supervisor, ***Dr. Siti Salwah Salim*** who has given me an opportunity to do this final year project. Thanks to her valuable of advice, support and guidance. I would like to thank ***Cik Nor Aniza***, the moderator of the final year project.

My special thanks to my fellow course mates, especially Chuan Yap Kang, Wong Siew Wai, Low Sai Kit and Ong Khian Guan for sharing their knowledge. To my friends Tan Su-Zan, How Siow Ting and Jessie Tan for their support and advice.

Finally, thanks to my family for being understanding, caring and supportive for the past few months.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Chapter 1 INTRODUCTION

1.1 Project Overview

Nowadays, it is common for an application to provide user interface. The tremendous changes and fast moving in software industry, it is useful that if a package is capable of providing numerous types of user interfaces features and graphics, so that the developer or user may concentrate on their scripting and save time for the interface invention.

Graphical user interfaces (GUI) provide a pictorial means for a user to interact with an application. An application represents its data using various types of graphical objects and the user interacts with the application by manipulating the properties of these graphical objects, such as menus, icons and buttons.

Traditionally, the developers using the features that provided in Lotus Notes to design Lotus Notes navigators, layout regions and forms. However the features provided are limited. Almost every interfaces that being designed are about the same. This makes the developed database interface not attractive as well as effective. Developers tend to find some other graphics elsewhere so that the designed user interfaces are more interesting and intuitive. The success or failure of an application often depends heavily on the ease of use of the application. Lotus Notes developers have faced problems on creating interesting GUI.

Hence, Notes GUI Gallery (NGG) is to provide a wide range of user interface graphics and menu templates, which would enable a clear presentation of the application. It's user friendly toolkit and designed to let the Notes developers to design or create a database.

NGG allows users to create more outstanding and interesting menus or interface easily and more effectively. Developers enable to use both graphics and menu template components easily by copying the selected ones and paste it on their application interfaces. This project focuses on the user interface toolkit, which is targeted at Lotus Notes developers by providing a library with graphics and menu templates.

This toolkit consists of two main components namely graphics and templates. Both components have the same functions, such as copy, add, delete and modify the graphics or templates. However the contents of both components are different.

Graphics component is a library contains of all types of graphics. The major elements in this component are lines, buttons, icons, boxes and pictures. There are different kind of lines, buttons, icons and pictures. Samples of content in graphics component are shown as below :

- Lines - curve lines, arrow lines, double lines, dashed lines, dotted lines and etc.
- Buttons - graphics buttons, command buttons and etc.
- Icons - colour icons, graphics icons and etc.
- Pictures - animals, nature, portraits, patents and etc.

Another component is a collection of templates for navigators and forms. The navigators and forms templates are meant for company workflow and collaboration learning space, which are very popular nowadays.

1.2 Project Objective

The aim of this project is to develop a graphics storage toolkit under Lotus Notes to let the Notes developer to design user interface and create database. To achieve this aim, objectives are set as below :

- To provide a wide range of graphics and templates for developers to create a more attractive user interfaces.

- Shorten the time for them to develop an application. Developers are no longer wasting too much time in inventing the user interface.
- Develop a toolkit on Notes with user friendly environment so that Lotus Notes developers can concentrate on their scripting. Developers don't need to spend much time to learn to use the toolkit.
- Develop for enhancement (add and delete).
- To reduce the complication of creating new database.
- To upgrade the conservative method of designing user interface and database.
- To simplify the designing and creating process.
- To reduce creating errors due to lower level of human intervention.

1.3 Project Scope

NGG have lots of features to cater the complicated of designing the interfaces in Lotus Notes. This project covers the scope as below :

- Providing different types of graphics, such as lines, boxes, icons, buttons and pictures as well as templates for navigators and forms.
- Users are able to add or delete new graphics and templates into the toolkit for their future usage.
- The selected graphics are movable on the designed interface, so that the users can put that anywhere.
- Users are able to collect their favourite or frequent used graphics and templates into their favourite folders so that they can easily retrieve them.
- When designing the interfaces, Lotus Notes developers are able to write Lotus script for each graphics to make it as functionality graphics.
- The graphics component of this toolkit can be used for other applications rather than Lotus Notes.

1.4 Significant of Project

This project is significant in a sense that it fulfils the needs of Notes developers in designing user interfaces for Notes application. Why choose Lotus Notes? Lotus Notes is a powerful GroupWare. It enables communication, collaboration and coordination among employees, customers and suppliers in business field; and students, teachers and administrators in education field. Notes applications include electronic mail, electronic forums or discussions, workflow automation, the management of quality, customer support, sales, support for mobile workers, and many other possibilities. Lotus Notes has been widely used in business and collaboration learning space.

"Together, Lotus Notes and Microsoft have about a 60 percent share in the client/server e-mail market, according to the newsletter. The remaining 40 percent is shared by Novell Inc.'s GroupWise, SoftArc Inc.'s FirstClass, Hewlett-Packard Co.'s OpenMail and Fujitsu Software Corp.'s TeamWare."[1]

However, Lotus Notes is not very powerful to support graphics user interfaces. Lotus Notes do not provide any graphics for forms, navigators, layout region and even database icon. The design components that are provided in Lotus Notes are hotspots, rich text field, command button, list boxes and combo boxes. None of them are graphical components. On the other hand, Lotus Notes provided the feature to import graphics through clipboard. Notes developers need to save the graphic they want to the clipboard of other windows applications, such as Microsoft Paint and Adobe PhotoShop.

With such features in Notes Lotus, Notes developers have to spend time in designing user interface. They will either download the graphics from other sources like the Internet or draw the graphics using painting software. This will take a long period of time for them to come out with all the graphics needed in developing an application. Sometimes they spend more time on designing user interfaces than developing the application.

In an application there is a lot of user interfaces. With no graphics toolkit, sometimes the developed application will have the similar design for every user interfaces if the Notes developers do not put an effort on it. Bad and unpersuasive user interfaces mean the system is bad and fail to fulfil the functional requirements. It is because user interfaces are the interaction between users and system. A bad interface will fail to let the user to understand the system. Therefore most of the Notes developers focus on their design in user interface, which is as important as develop an application.

With such significance, this project directly helps any Notes developers in developing a Notes application.

1.5 Project Outcomes

NGG focuses on developing a design library that can be used by Notes developers. NGG have the following characteristics:

- An interesting database that consists of all types of graphics and templates.
- A GUI library that fulfill the needs of Lotus Notes developers meanwhile solving the problems of having limited graphics in Lotus Notes.
- User friendly toolkit, which means it is easy to use.
- This toolkit is a stand-alone application that can be used or accessed by Lotus Notes.
- Able to provide a folder to keep the entire favourite graphics or templates.
- It has a potential usage for Notes development.
- The usage of NGG could be implemented not only for Lotus Notes but also the other windows applications.

1.6 Project Development Strategy

For this project, prototyping model is selected as a process model in the software engineering literature. Prototyping model uses iterative approach where each system development life cycle (SDLC) is repeated several times (iterated). Requirements and alternative solutions are analyzed in the process. During this process, solutions are also designed and some portion of the system is implemented and subject to a user review. Prototyping model consists of six steps as shown in Figure 1.1.

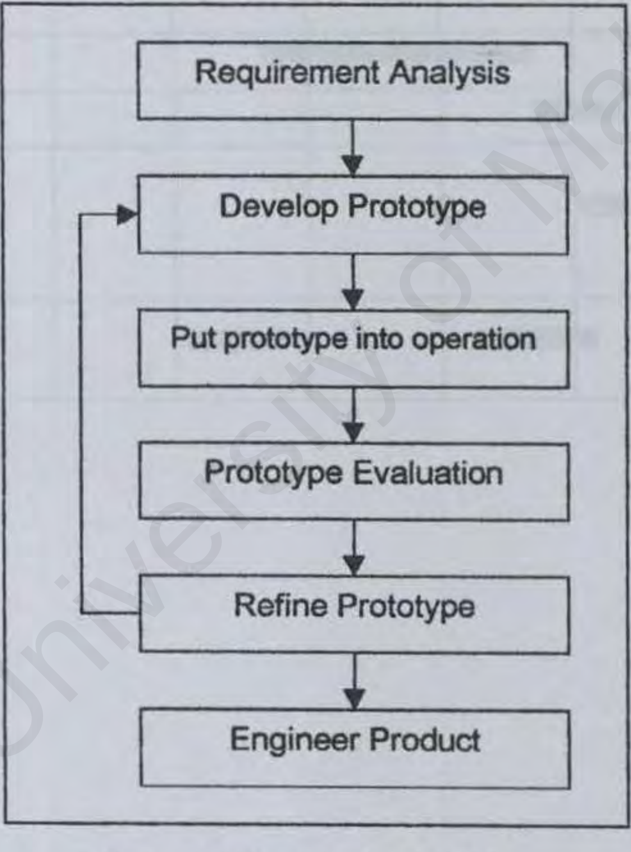


Figure 1.1 Prototyping Model

1.7 Project Schedule

A project schedule describes the software development cycle for a particular project by enumerating the phases or stages of a project and breaking each into discrete tasks or activities to be done. The schedule of this project is shown in Gantt chart below.

Task/Activity	Jun'99	Jul'99	Aug'99	Sep'99	Oct'99	Nov'99	Dec'99	Jan'00
Research	█	█						
System Analysis			█					
System Design			█	█	█			
Development					█	█	█	
System Implementation Testing						█	█	
System Documentation			█		█			█

CHAPTER 2

LITERATURE REVIEW

Chapter 2 LITERATURE REVIEW

The review covers the current toolkits that are used in Lotes development environment to build better Graphics User Interface and databases.

2.1 What is Toolkit (Tk)?

Toolkit is a companion program to Tool Command Language (Tcl) for creating graphical user interfaces. Together with Tcl, Tk is a rapid program development tool from Sun Microsystems.[2]

Toolkit provides a collection of standard buttons, menus, listboxes, and scrollbars and two special features, the text widget and the canvas widget, that the Tcl programmer can use to speed up program development.

2.2 Interactive System

Interaction with computer involves at least two participants, which are the user and the system. Both are complex, and are very different from each other in the way that they communicate and view the domain and the task. The interface must therefore effectively translate between them to allow the interaction to be successful.

The purpose of an interactive system is to aid a user in accomplishing goals from some application domains. Interactive system as its name indicates, an interactive system supports communications in both directions, from user to computer and back. It does this in a way that enables it to follow the pace and direction of the user's activity. The user takes actions such as pressing buttons, pointing with a mouse or typing in text. The system reacts accordingly, perhaps by displaying information, perhaps by activating machinery or performing some other useful service, perhaps just by waiting for the user's next action. All of this takes place via the system's user interface, the part of the system that provides access to the computer' internal sources.

In today's world, interacting with computer-based devices and system is common place. Even home TV set and wristwatch contain large amounts of software. Interacting with computers, once just a privilege of the professional and clerical workforce, is now something that everybody does.[3]

2.3 User Interface

User interface is one of the important structured approaches in any applications especially software applications. The user interface is the hardware and software that facilitate communications and interaction between the user and the computer. User interfaces are a subset of the field called *human-computer interaction* (HCI). Approach to human-computer interaction design and evaluation has the goal of systematizing these processes. Actually what is human computer interface?

2.3.1 HCI (human-computer interaction)

HCI (human-computer interaction) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings. A significant number of major corporations and academic institutions now study HCI. Historically and with some exceptions, computer system developers have not paid much attention to computer ease-of-use. Many computer users today would argue that computer makers are still not paying enough attention to making their products "user-friendly." However, computer system developers might argue that computers are extremely complex products to design and make and that the demand for the services that a computer can provide has always outdriven the demand for ease-of-use.

One important HCI factor is that different users form different conceptions or mental models about their interactions and have different ways of learning and keeping knowledge and skills (different "cognitive styles" as in, for example, "left-brained" and "right-brained" people). In addition, cultural and national differences play a part. Another consideration in studying or designing HCI is that user interface technology changes rapidly, offering new interaction possibilities to which previous research. Findings may not apply. Finally, user preferences change as they gradually master new interfaces.[2]

2.4 Graphical User Interfaces

2.4.1 Introduction

A Graphics User Interface is a graphical (rather than purely textual) user interface to a computer. The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands that need to be remembered and computer responses that were infamously brief. The command interface of the DOS operating system (which still can be get from Windows operating system) is an example of the typical user-computer interface before GUIs arrived. An intermediate step in user interfaces between the command line interface and the GUI was the non-graphical menu-based interface, which interacted by using a mouse rather than by having to type in keyboard commands.[2]

A study has shown that users in a GUI environment work faster, more accurately and with lower frustration and fatigue level the users in a character-based environment. Productivity has increased 35%, and the accuracy of completed work has improved as much as 74% in some environment.[4]

2.4.2 GUI Development

The first graphical user interface was designed by Xerox Corporation's Palo Alto Research Center in the 1970s, but it was not until the 1980s and the emergence of the Apple Macintosh that graphical user interfaces became popular. One reason for their slow acceptance was the fact that they require considerable CPU power and a high quality monitor, which until recently were prohibitively expensive.

Many DOS programs include some features of GUIs, such as menus, but are not graphics based. Such interfaces are sometimes called graphical character-based user interfaces to distinguish them from true GUIs. Graphical user interfaces, such as Microsoft Windows and the one used by the Apple Macintosh, feature the following basic components [5]:

- **Pointer** : A symbol that appears on the display screen and that user move to select objects and commands. Usually, the pointer appears as a small angled arrow.
- **Pointing device** : A device, such as a mouse or trackball, which enables users to select objects on the display screen.
- **Icons** : Small pictures that represent commands, files, or windows. By moving the pointer to the icon and pressing a mouse button, users can execute a command or Convert the icon into a window. Users can also move the icons around the display screen as if they were real objects on user's desk.
- **Desktop** : The area on the display screen where icons are grouped is often referred to as the desktop because the icons are intended to represent real objects on a real desktop.
- **Windows**: users can divide the screen into different areas. In each window, users can run a different program or display a different file. Users can move windows around the display screen, and change their shape and size at will.
- **Menus**: Most graphical user interfaces let users execute commands by selecting a choice from a menu.

2.4.3 Advantages of GUI

The advantages of GUI are as below :

- The icon can take up less screen space than the corresponding text description of the functions.
- It can be understood more quickly if well designed, and can initiate a whole series of operations or activities.
- The truly unique benefit provided by a windowing system is the ability to have multiple views of different objects on the screen at the same time.
- It has the capabilities to make the program easier to use.
- Well-designed graphical user interfaces can free the user from learning complex command languages. Many users find that they work more effectively with a command-driven interface, especially if they already know the command language.
- A common GUI provides a friendlier, less intimidating and less confusing environment. A GUI environment offers familiar symbols or icons that resemble commonly known functions or items such as files and trashcans so that complicated, arcane command-line sequenced do not have to be memorized.
- Users are definitely attracted to GUIs because of the promise of common interfaces across programs and ultimately across platforms, and an expectation that the user will have more control over the computer.
- It is easy to move data from one application to another. A true GUI includes standard formats for representing text and graphics. Different programs that run under a common GUI can share data. For example, to copy a graph created by a spreadsheet program into a document created by a word processor.

2.4.4 Principles of designing GUI

A design principle is a general statement that describes some aspect of user interface design 'best practice.' These principles usually have a basis in research concerning how people learn and work. Good user interfaces conform to good design principles. The following design principles are applicable to all user interfaces [6]:

Design Principle 1: Maintain consistency at all levels

Consistency is the primary reason for design and style guidelines. Consistency is important both within a single product and across a product line (or enterprise). Users become habituated to specific actions and expect consistency. If actions are executed in a specific way on one screen, users expect the action to be performed in the same way throughout the application. They expect certain features of an interface to behave in certain ways, and can become confused or frustrated by unexpected behavior.

Consistency is also required in data presentation, data entry, labeling, text style (including capitalization and fonts), and in terminology. In addition, by making the interface familiar and predictable, consistency allows users to transfer existing knowledge to new tasks. The user can learn new things more quickly, and is able to focus on the task rather than the tool.

Areas where consistency is important include (but are not limited to): abbreviations, function keys, accelerators, icons, acronyms, keyboard navigation, alarms and warnings, message location, color meanings, numeric data display, currency, date and time format, data entry prompt, terminology.

Design Principle 2: Minimize demands on human memory

One of the most common errors in user interface design is to rely on the user's memory about how to interact with the program. There are a several ways in which this can occur.

Cognitive scientists generally describe two types of memory: short-term and long-term. Short-term memory, or working memory, is like a buffer. It holds limited information, which fades quickly, unless it is refreshed. Research has shown that most users can hold only about 4 to 7 discrete chunks of information in short-term memory. Working within this limitation leads to designs that:

- limit the number of items a user has to deal with at any given moment;
- don't require the user to 'buffer' information (on paper or in memory) from one screen to the next;
- allow users to recognize rather than recall;
- Facilitate task closure (use short, linear action sequences).

Long-term memory is the user's conceptual storage. Long term memory holds the concepts, which make an interaction meaningful. In terms of the user interface, there are several functions of long-term memory:

- To store the basic technical concepts which the user possesses. For example, a user's understanding of such terms as "file" and "database" depend on the concepts in his or her long-term memory.
- To store basic domain information which relates to the program, where 'domain information' refers to aspects of the job.
- To store a mental model (or cognitive map) of the software. A mental model of how the software operates guides the user in its operation. Users may posse more or less complete mental models depending upon their technical skill and familiarity with the software. Mental models are built up over time, and are based on experience with the software product.

Design Principle 3: Present informative feedback

Feedback closes the communication loop between the computer and the user. Feedback tells the user that their actions have been processed and what the results of those actions are. In the absence of error messages, normal feedback lets the user know that the system is behaving in the expected manner. As an example, users typically become uncomfortable when there is no indication of activity between the initiation of a command and its execution. The use of feedback in the form of a unique cursor shape or message assures the user that the command is being processed, and the system hasn't crashed.

Two types of feedback are typically needed. Articulator feedback tells the user that his or hands worked properly. For example, "when the user selects 'open' from a pull-down menu, the choice may blink briefly just before the menu disappears. This is articulator feedback indicating to users that they picked their intended choice" [7]. Semantic feedback tells the user that his or her head worked correctly. In the example above, a dialog box listing files that can be opened appears; this confirms that the user selected the appropriate task.

Feedback takes many forms in the user interface: visual clues, like cursor shape and animation; audio clues, like alarms; and messages. Messages can be used to provide guidance, to warn users of potentially dangerous actions, or to confirm commands. Messages should appear in consistent locations, and should be clearly written in the user's terms.

Design Principle 4: Keep the user in control

In general, the user—not the computer—should initiate actions. The locus of control has a tremendous psychological impact and strongly influences the user's feelings about an interface. It is important that the user feels that he or she is in charge, not the computer.

"The user should have the impression that the computer is prepared to respond whenever the user is ready to issue a command. Subtle differences in wording of messages can convey this in a surprisingly forceful way. Think about the difference between 'enter next command' and 'ready for next command.' 'Enter next command' is a very demanding message from the computer, to which the user may feel pressure to take action. However, 'ready for next command' is a message that implies to the user that the computer is prepared to respond on demand from the user." [7]

User control can be reinforced by allowing the user to customize the interface, and accommodating different modes of interaction (pointer, keyboard).

Accommodate users with different levels of experience. A well-known usability expert tells the story of a client who complained that an interface was too easy to use—it 'coddled the user.' On further discussion of this complaint, it became clear that the critic was a power user who didn't want to be slowed down by the interface. The design team responded by designing a path for power users which bypassed some of the structures developed for less proficient users. The resulting design satisfied both inexperienced users and the critical power user.
The use of modes affects locus of control. When modality is used, make sure the user can distinguish which mode is active. For example, graphics software will often use different cursor shapes to distinguish which mode is active.

Avoid the use of preemptive modes (those requiring a user to complete one task before continuing to another) wherever possible.

Design Principle 5: Use a consistent conceptual model

The use of a consistent conceptual model for the interface helps provide coherence. It also allows the user to develop a consistent mental model of the system. The absence of a well-defined conceptual model may lead to the use of randomly added elements, and as a result, user confusion.

Designers use several types of conceptual models and metaphors: visual, functional and stylistic. Perhaps the best-known visual metaphor is the desktop metaphor developed by Xerox, which evolved into the interfaces used by the Apple Macintosh and Microsoft Windows. VisiCalc, the first electronic spreadsheet, was modeled on an accountant's financial form.

Using a conceptual model has several advantages:

- It provides a form of structural integrity for the design process. If the interface is modeled on a well-defined concept or metaphor, the designers can test each element to see if it fits the model. In interfaces, as in the real world, objects are rarely used in isolation. A table setting includes plates, glasses, napkins and silverware. While it is reasonable to add a pair of chopsticks to the place setting, adding a hammer is not—because a hammer violates our concept of tableware. In interactive designs, the conceptual model can be used to check the consistency of objects and actions. For example, a trashcan fits the desktop metaphor, while a dump truck does not.

- It helps the user develop a mental map of the software. For example, many musicians now use computers to record and edit their material. Although MIDI (Musical Instrument Digital Interface) and digital sound recording systems have little in common with traditional tape recorders, most programs use an interface modeled on the controls of a tape recorder. Thus, the musician can think in such terms as "rewind," "playback" and "record" rather than technical terms.

- It facilitates the use of direct manipulation. Unlike indirect manipulation (the user specifies an object in one action, then specifies an operation to be performed on the object using a separate action), direct manipulation allows the user to specify an object using a mouse and perform an operation by means of a physical mouse action.

Sometimes a conceptual model or metaphor will be obvious; at other times, the design team may struggle to find one. It is quite possible to design elegant usable systems which have no special metaphor underlying them at all. However, all well-designed systems will be based on a conceptual model, which can be articulated, reviewed and refined.

Design Principle 6: Design for prevention of and recovery from error

Users learn by trial and error. An effective interface provides only appropriate choices, warns users of potential problems, and allows users to backtrack by providing 'undo' and 'backup' functions.

One technique that helps users make appropriate choices is disabling or 'graying out' menu choices or buttons when they are not available. While this approach has the advantage of protecting the user from erroneous choices, it makes the designer's job more complicated, and may frustrate a user who does not understand why a choice is unavailable.

Designers should also require confirmation of potentially destructive actions. Use a message that clearly warns the user of the consequences of the requested action. Ask the user if this is what he or she intended, and provide a way to cancel the command.

Design Principle 7: Design for simplicity

Simplicity is the absence or reduction of complexity. Several types of simplicity contribute to a well-designed user interface:

- Visual simplicity is achieved by showing only the most important controls and objects. Screen layout should follow good visual design practices. Use white space as a visual element to define perceptual areas.
- Verbal simplicity comes from the use of direct, active, positive language.

- Task simplicity is achieved when related tasks are grouped together, and only a few choices are offered at any one time.

Conceptual simplicity is accomplished by using natural mappings and semantics, and by using progressive disclosure.

Design Principle 8: Design the software to mirror workflow

Procedures implemented in software should always reflect the task structure of the work they are supporting. This requires analysis of the user's tasks and creation of software models to reflect them.

2.5 Lotus Domino and Notes

2.5.1 Introduction

Lotus Notes is a GroupWare technology from Lotus Development Corporations. GroupWare enables a group of people in an organization to use the same information in various ways that fit their specific and individual needs. It can enhance decision-making, and it can encourage the spread of ideas and critical thinking. It support communications, collaboration and coordination between people work in a groups.[8]

The most important component of any Notes application is the Notes database, which is an unstructured database as opposed to relational database. Though it's database Notes offer e-mail, calendar and scheduling capabilities.

Currently, Notes software is divided into the server component and the client component. The client component is still called Notes whereas the server is called, Domino. Apart from acting as a Notes server, Domino 4.6 also acts as a web server and is able of translating Notes database into HTML format dynamically.

Notes provides the following:

- Email
- Group discussion
- Workflow
- Document management
- Scheduling
- Application development
- Web publishing and browsing
- Distributed document replication
- Centralized directory services(address book)
- The ability to integrate with disparate email systems (Notes offer SMTP, MTA, cc:Mail, and POP3 support, for example)

2.5.2 What is Lotus Notes good for?

Business benefits greatly not only from Notes' powerful email capability but also from the ability to redefine and automate their business processes. For example, some business have used Notes to automate their hiring or purchase approval process to include signature authorizations at each phase. Some companies run all their business communications on Notes, often using Notes as a front-end data-gathering tool for information that eventually ends up stored on a mainframe. Many companies also build reporting applications in Notes that import data from mainframe computers to report across a wide range of users in corporation.

2.6 Design Features in Lotus Notes

Designing forms and views in Notes entails more that just defining the static text and fields to be used. There is an aesthetic component to designing a compelling user application as well as the functional component. Sometimes this aesthetic aspect is called the 'fit and finish' of a product; sometimes it is called the 'form' of a product as opposed to the product's 'function'.

A product, whether it is an automobile or a computer application, must function correctly and perform well, but its look and feel is important, too. The way the application looks makes a difference in how it is perceived by users as well as affecting its usability. It may even affect how well the users accept and like using the application.

In designing an application, it should be almost automatic for the user. The user should be able to concentrate the business at hand. The interface should look good and work well. The user should be able to perform the business task without even thinking about the user interface. Many of the usability enhancements available in Notes 4.6 of Notes can improve the visual appeal of the application as well.

2.6.1 Design elements in the database

Notes have six main design elements [9]:

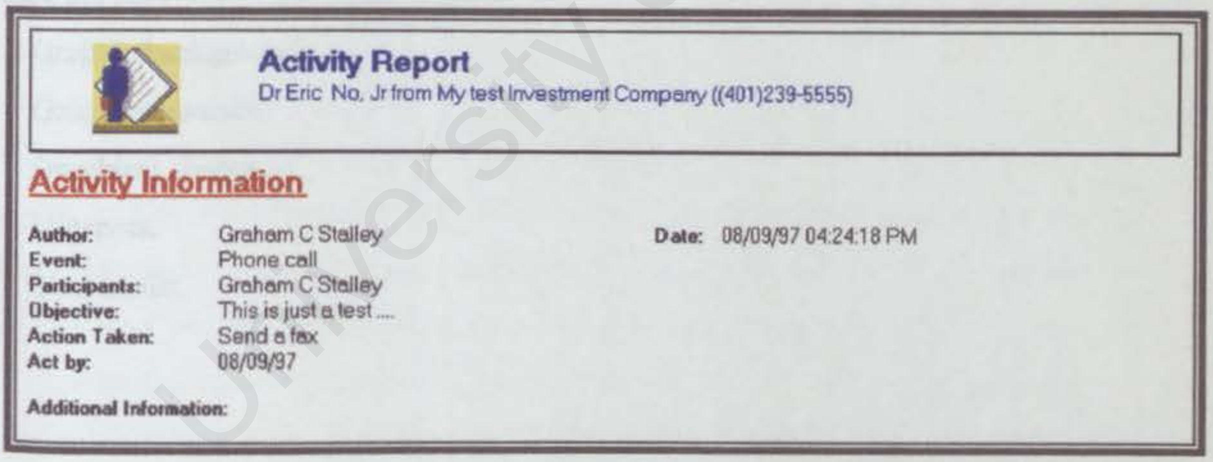
- **Forms** define the data that will be entered into document. They consist of static text, fields, graphics, buttons, subforms, sections and actions. Forms are the hearts of an application. They are the primary user interface for entering data into the database and contain many of the actions that users invoke while using the application. However, user data is stored in documents. Forms simply provide a way to display and capture the data; they typically aren't a part of the document itself.

- **Views** are a list of the documents contained in the database that display specific information about each documents using rows and columns to delineate the information. Views are indexes to the information in the databases. A single database usually has many views that list different subsets of documents or different information or sort the documents in different orders or categories. A well-designed set of views makes information in the databases easy to find and more useful
- **Subforms** are reusable groups of fields, graphics, and other form design elements. They are somewhat of a form within a form. Subforms can include many of the same elements that are placed on forms. By using them, users don't have to add design elements to each form individually; users can just insert the subform. Also, if users need to make changes to the subform, users can make the changes once – in the subform – and they will propagate to all forms using the subforms.
- **Navigators** are graphical interfaces to the databases. They can contain icons, buttons, bitmaps, hotspots, and static text. Clicking an element on a navigator can execute Simple Actions (such as opening a view or another navigator, @Function formulas, or LotusScript programs. Navigators make the applications more intuitive and easier for the users.
- **Script Libraries** allow Notes developers to build reusable LotusScript modules that can be invoked from multiple places within a Notes database.
- **Agents** are programs written in either LotusScript or the formula language using @Functions and @Commands. Users can run then a predetermined interval from a menu or event. Agents can be either private or shared. Some good uses for the agents include archiving a set of documents, categorizing documents into folders, updating groups of documents, or sorting common tasks that might be performed multiple locations within a database.

2.6.2 Forms/ subforms

Forms define the format and layout of documents. A form contains several design elements that help define the structure of the database. When a form is designed, these objects can be placed anywhere. Here are the most widely used components for designing forms:

- Fields
- Text
- Graphics
- Hotspots
- Action bars
- Tables
- Layout regions
- Attachments
- Links
- Java applets
- OLE objects
- Graphic background
- Background color
- Blank lines
- Others



Activity Report

Dr Eric No. Jr from My test Investment Company ((401)239-5555)

Activity Information

Author:	Graham C Stalley	**Date:**	08/09/97 04:24:18 PM
Event:	Phone call		
Participants:	Graham C Stalley		
Objective:	This is just a test		
Action Taken:	Send a fax		
Act by:	08/09/97		

Additional Information:

Figure 2.1 Sample of Form

2.6.3 Navigators

As an application is built, however, it's good idea to implement any navigators with buttons only, to prove the concept. Then return later and have a person specializing in graphic arts design create useful artwork for them.

Navigators can occupy the entire screen, or they can appear in the pane normally occupied by the views and folders outline. Experiment with both of these and realize that where users want to drag documents into a folder on the navigator. [9]

Once forms and views for database is created, a user interface that helps users maneuver through it, will be built. Graphical navigators are that interface, similar to views. Navigators provide a graphical way for users to do such things as switch views, open documents, file documents into folders, and just about any other action can be programmed in Notes.

Navigators can made up of objects (text, pictures, or shapes) that cause actions to occur when they are clicked. One common use of navigators is a graphical table of contents. Even though the navigator provides a good interface to a database, users undoubtedly will want to create their own specialized navigators

There are six types of navigator objects:

- Graphic backgrounds.
- Graphics buttons.
- Graphical shapes.
- Hotspots.
- Text boxes.

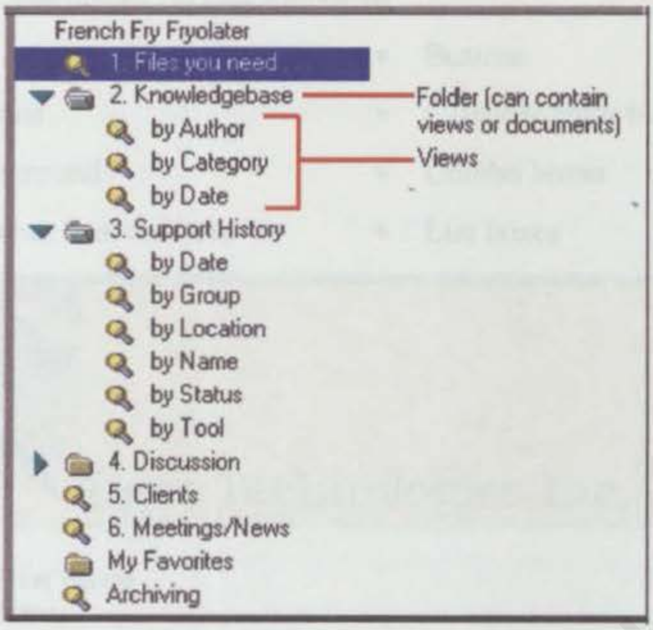


Figure 2.2 Sample of navigator

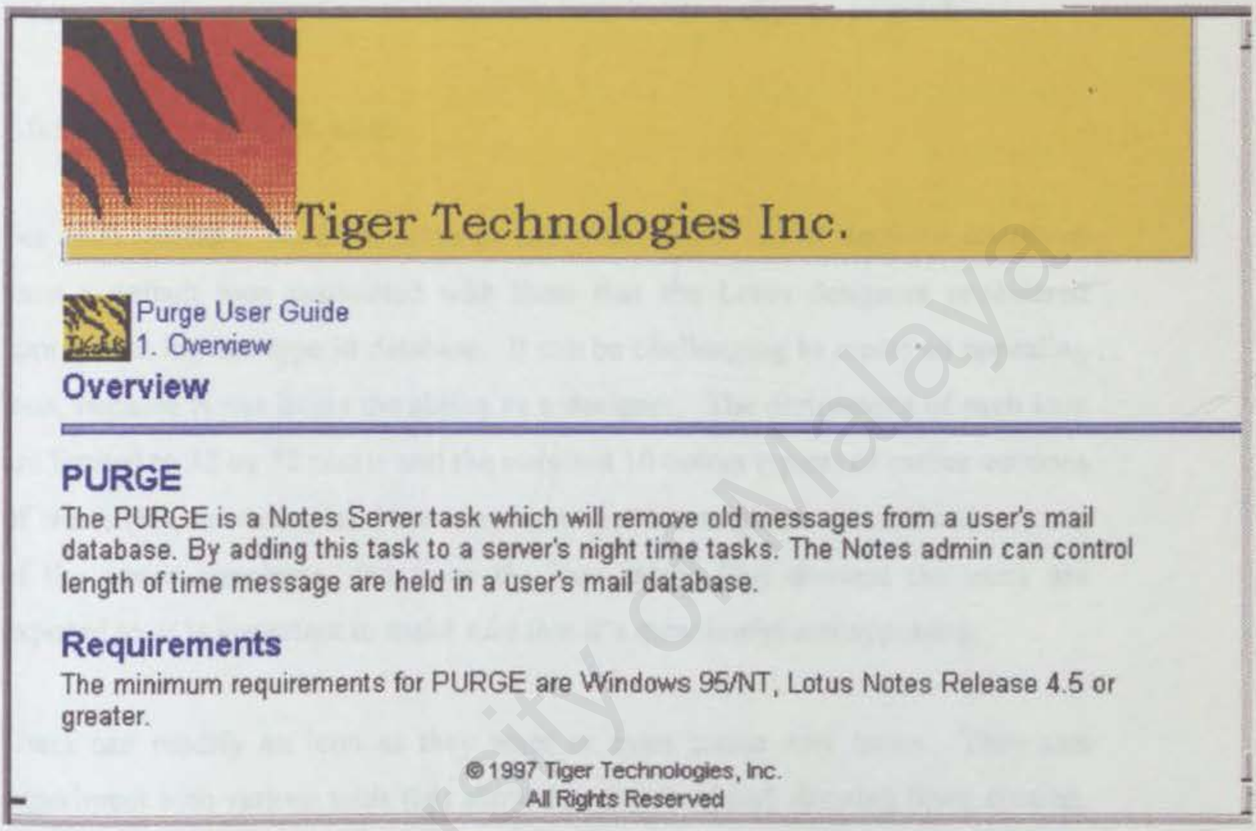
2.6.4 Other design elements

2.6.4.1 Layout regions

Layout regions are designed to help users to lay out their forms. Layout regions allow predetermining the vertical and horizontal dimensions of a field and enable scroll boxes, multilines fields and so on. Layout regions are fixed in size and position on the form. Also, within a layout region users can easily drag and drop the fields and text to get just the look they want. However, layout regions don't support rich text, shared fields, subforms, computed text, and other robust features. Layout regions aren't supported on the Web. However, when designing applications for native Notes, layout regions facilitate the design of powerful and friendly user interfaces.

Design elements that layout regions supported :

- Static text
- Graphic buttons
- Graphic background
- Field types other than rich text
- Buttons
- Calendar entry buttons
- Combo boxes
- List boxes


Tiger Technologies Inc.

Purge User Guide
1. Overview

Overview

PURGE

The PURGE is a Notes Server task which will remove old messages from a user's mail database. By adding this task to a server's night time tasks. The Notes admin can control length of time message are held in a user's mail database.

Requirements

The minimum requirements for PURGE are Windows 95/NT, Lotus Notes Release 4.5 or greater.

© 1997 Tiger Technologies, Inc.
All Rights Reserved

Figure 2.3 Sample of layout region

The following types of fields are not allowed to a layout region:

- Attachments.
- Hotspots
- Links
- Objects
- Tables
- Popup
- Rich text fields
- Sections

2.6.4.2 Tables

Tables are useful for summarizing information or lining up fields in rows and columns. A table placed on a form appears in every document created with the form. Tables can be used to organize information or line up fields' rows and columns. Tables within forms can contain text, buttons, objects, or graphics.

2.6.4.3 Database icons

For each database, Notes displays an associated icon. Most database templates have a default icon associated with them that the Lotus designers considered appropriate for that type id database. It can be challenging to create an appealing icon, because Notes limits the ability as a designer. The dimensions of each icon are limited to 32 by 32 pixels and the standard 16 colors typical of earlier versions of Notes (R3, for example). However, some good examples are contained in some of the newer templates, and since the icon is the first element the users are exposed to, it is important to make sure that it's meaningful and appealing.

Users can modify an icon as they want or even create new icons. They can experiment with various tools that allow drawing freehand, drawing lines, erasing, filling large area of the icon with certain colors, and perform other special editing. The users can also use icons from other sources. They might also have access to an icon library database that contains all kind of icons that others have created, and users can use them for their database.

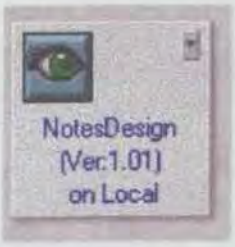


Figure 2.4 Sample of database icon

2.6.5 Design components

2.6.5.1 Graphic backgrounds

A graphic background is a bitmap that is pasted into the navigator. Each navigator has only one background, and actions can't be attached to it. In fact, a graphic background is the only element that can't perform an action. It is similar to wallpaper behind the information that appears on-screen. Use a Graphic Background if users have at their disposal an image that contains areas that the user would immediately identify as discrete objects that can be selected to initiate an action.

If a graphic background is used, first a bitmap is pasted to the clipboard from another application. Notes does not enable bitmapped files to be loaded directly from disk. Graphical background can't be resized once they are pasted into a navigator.

2.6.5.2 Graphic buttons

These are small images that can be pasted into navigator. They can appear like icons on-screen and perform specific actions when clicked. Graphic buttons can't be transparent, and any caption must already be part of the graphics.

Figure 2.5 Sample of button

2.6.5.3 Graphical shapes

Similar to buttons, graphical shapes can be rectangles, polygons, polylines, or ellipses. They are drawn using Notes drawing tools and can be any shape. Like graphic buttons, Notes tasks can be assigned to perform when hey are clicked. Graphical shapes can be transparent and can have captions.

2.6.5.4 Hotspots

Hotspots are very similar to graphical shapes, except that they are displayed on-screen in more discreet manner. For example, a hotspot might be a green pop-up box that appears around text. When it's clicked, additional information might appear. They are transparent so that they don't take up a lot room on-screen. Hotspots are always transparent and don't have captions. Hotspots, either in the form of a rectangle or a polygon. A hotspot is simply an outline that can be used to identify a particular region of a graphic background. Because hotspots are only outlines, they are never filled with a color. Hotspots can be used, for example, to outline geographical regions, such as cities, on a state map.

2.6.5.5 Text boxes

Text boxes are simply blocks of text that can be placed on the navigator and that can have actions associated with them. Text boxes can be transparent. Of course, they have captions, since they are text boxes. Font, size and color are all configurable.

2.6.5.6 Command buttons

Command buttons ate normal buttons with a text caption on their face. Command buttons are useful for initiating any actions that don't have graphical descriptions. Command buttons can't be transparent and can have captions.

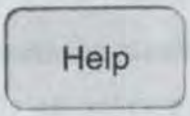


Figure 2.6 Sample of button

2.6.5.7 Objects

For objects that will trigger events, one or both the options presented in this screen should be selected:

- Highlight When Touched. Enabling these options will draw the border for the object, as specified by the Highlight Outline Width and Highlight Outline Color settings, whenever the user moves the mouse over the navigator's object. This option should typically be selected for hotspot polygons or hotspot rectangles.
- Highlight When Clicked. Enabling this option will flash the border for the object, as specified by the Highlight Outline Width and Highlight Outline Color settings, when the user clicks on the navigator objects.

2.6.6 Templates

Notes come with a collection of templates – ready-made databases – which can be used as a starting point for creating a new database. A template is a special type of Notes database (with a .NTF extension) from which other database are modeled. Templates contain no data, only design elements that can be used to serve as a guideline when start creating a new database. This will minimize the development time since these databases already have various forms, views, agents, navigators, script libraries, and other design elements. When creating a new database, it can also just as efficient to copy an existing database and modify the design where appropriate.

If a template is used, the first step in creating a database is deciding which templates should serve as the starting point for a new database. In some cases, one of the supplied templates might meets the functional requirements. If none of the templates matches the needs exactly, an existing template still might contain many of the design elements that is required and therefore can serve as a starting point for creating a new database.

2.6.6.1 Reviewing the Notes Templates

Notes come with 44 templates (including advanced ones). A few in particulars are extremely useful and can be used to create new databases, including powerful Web pages, with little or no customization on developer part. If developers design new database often, they might come back to these as starting points or create their own templates with standard views, forms, subforms, agents, script libraries, and so on, thus minimizing development time. Notes templates are fully functional databases that can be used "off –the-shelf". However, because no two business operate in the same manner using the exact same processes, the applications might need to be customized to meet the needs of the organization. Nevertheless, in these cases, templates can still be utilized as starting points for database design. [8]

2.6.6.2 Some of the standard templates available with Notes

Templates Name	Description
Discussion Notes and Web (R4.6)	Discussion databases are databases in which workgroup can share thoughts, ideas or knowledge. Team members can read and contribute to discussion threads in which the history of each topic is preserved. This template is designed to be accessed from both Notes client and Web browsers and is fully functional for both types of clients.
Document Library (R4)	It is used for storing and describing documents much like an electronic filling cabinet. Typically, this serves as an object store and allows for the set-up document approval cycles and archiving
Personal Journal (R4)	It lets users keep a diary or journal and categorize entries into folders. This also contains "clear sheets": and serves as a scaled-

	down word processor with versioning control.
Personal Web Navigator (R4.6)	It gives user accesses to the Internet directly from the Notes workspace using Notes as the browser or Internet Explorer within Notes.
Search Site	It allows users to search multiple databases using only one search.

2.7 Existing Toolkits

2.7.1 RADD GUI Gallery

RADD GUI Gallery is a library of professional 3-D graphics that can be copied and pasted into Notes Release 4 design elements such as Navigators, layout regions, and forms. The images are targeted toward the more conservative corporate environment where extravagant World Wide Wed graphics are usually in an appropriate. [10]

Some of the features that are available:

- A collection of templates for Navigators, forms, and dialog boxes.
- Using images - To use an image, simply click the Copy button next to the Image label on an Entry document. The image is now on the Clipboard and can be pasted into the application or into Windows Paintbrush to modify it.
- Entering images - To enter images, simply select creates entry from the Notes menu bar and completes the form that appears. The Copy button will copy everything in the Image field. Therefore, only a single image should be included in one field.
- Keeping track of favourite or most used items - To keep collection of the favourite or most used images, simply drag and drop any entry from a view to the Favourite folder on the Main Navigator. To remove entries from the

Favourite folder, highlight the entries in the folder and select Actions > Removes From Folder.

Advantages :

- Assist developers who may not have the time or inclination to work with the complexities of Notes R4 graphics.
- Allow developers t concentrate on coding their applications and still produce visually appealing interfaces.
- Enhanced graphics capabilities.
- Quickest and easiest way to create new database with templates.

Disadvantages :

- Limited graphics and templates.
- Provided graphics are not attractive.
- Not organised.

2.7.2 Troop Technology - Sources and Solutions Library for Lotus Notes

Sources and solutions library consists of a wide range of solutions in developing Lotus Notes application. Troop Technology searches many books, web sites, and help files for information on the ways to get things done in Lotus Notes. [11]

Some of the features that are available :

- A library of over 1000 examples of code and LotusScript for Domino, Java, Views, Forms, Agents, Selection Formulas, Translation, ODBC, OLE, and Validation.
- 2500 graphics, such as bitmap file and column icon chart.

- Web hints.
- All information for searching.
- Get them by just clicking a COPY button.

Advantages :

- It helps in figuring out a way to do something in formula or Lotus Script when developing Lotus application.
- Assist developers who may not have the time or inclination to work with the complexities of Notes R4 graphics.
- Developers don't have to memorize any formula or script when coding their applications.
- Quickest and easiest way to develop new Lotus application with interesting user interface and powerful coding.

Disadvantages :

- User unable to add and delete any features in this application.
- Difficult to get the solution because it contains too many of them.
- Descriptions are not clear.

2.8 Summary

Research had been carried out on available system. This step is particular important as implementing a toolkit has vast different comparing to existing feature. Many approaches and logic need to be redefined and reengineer to suit the requirements.

Due to rapid growth of Information Technology, graphic user interface toolkit is a favourite to design user interface in order to give away information effectively. However, to develop the GUI toolkit, there are some criteria that need to follow. For example, how does a graphic can be pasted on the user interface?

User interface toolkit has a lot of benefits. For example, users need not have to seek around web or draw on their own the graphics. Users just have to choose one from the GUI toolkit. However, each user interface system will have their restriction, for example, is it possible to import the graphics to the user interface?

With the knowledge in user interface, toolkit and Lotus Notes, a better user interface toolkit for the Notes application can be developed. This is the aim in literature review.

CHAPTER 3

SYSTEM ANALYSIS

Chapter 3 SYSTEM ANALYSIS

Systems analysis is the study of a current business and information system application and the definition of user requirements and priorities for a new or improved application. System analysis is driven by system users' concerns; hence, it addresses the PEOPLE, DATA, ACTIVITIES and network building blocks from a user perspective [12]. The overall emphasis of analysis is to gather data and to feasibility of these solutions. A complete understanding of software requirement is essential to the success of a software development effort.

System analysis is conducted with the following objectives in mind [13]:

- Identify the user's need.
- Evaluate the system-concept for feasibility.
- Allocate functions to hardware, software, people, database and other system elements.
- Establish time and schedule constraints.
- Create a system definition that forms the foundation for all subsequent engineering work.

3.1 Data Collection

Two methods of data collection are used to get and understand the functional requirements. The first method is by asking questions and the other method is by observational studies. Besides that, surfing through the Internet is also another method that is used to get the information.

3.1.1 Gathering information by asking questions

This is the first approach to be used to get the functional requirements. In this method, questions about the scope and functions of NGG were appointed to the project supervisor, Dr. Siti Salwah. A sample of an existing toolkit and answers to those questions are very useful in setting the functional requirements. Friends who have used Lotus Notes to develop applications before, were asked about what are the constraints they face when they are designing user interface and what are the features they hope to get in a graphics toolkit. A list of questions that have been asked to the Notes users is included in Appendix A.

3.1.2 Gathering information by participation

By asking question method can't really give a real picture of how the system should functions. Therefore another method is used to strengthen the information of functional requirements. This method is by participating into the system. Some graphical user interfaces were designed by using the features provided in Lotus Notes. This has given the information of what type of user interface that can be designed using only Lotus Notes and none other out-sources graphics. This type of user interface is plain user interface without graphics because there is no graphics provided in Lotus Notes. Samples of plain user interface that can be seen in Appendix B, contain only coloured background and command button.

Besides that, the sample of the existing toolkit, which is given by Dr. Siti Salwah, and some other Internet downloaded existing toolkits are used to design user interfaces. Those toolkits have given the information of functional requirements. Samples of user interface, which is designed using toolkit, can be referred in Appendix C.

3.2 Analysis of The System

Requirements analysis enable the system engineer to specify software function and performance, indicates software interface with other system elements and establishes design constraints that the software must meet. An analysis had been carried out on current available features in Lotus Notes and other existing toolkits.

In Lotus Notes, only some of the design elements are allowed to import graphics from other sources. There are forms, subforms, navigators, layout regions, database icons and tables. There are design elements can't import out sources graphics, such as actions button and hotspots. However templates can be created for forms, subforms and navigators.

An analysis is carried out on available existing toolkits. In the existing toolkits, they provide several of graphics and templates. The functions that existed in those toolkits are copy function and help function. In certain toolkits, they provide more features such as favourite folders. Ideas have been initiated after analysing those toolkits. NGG have more features, for example add and delete the unwanted graphics or templates.

Finally, the conclusion is a combination of ideas and information was obtained to build NGG. Entities, data are modeled, after the existing and current toolkit.

3.3 Consideration on the Development Tool

NGG is a graphical storage toolkit that contain all sorts of graphics as well as templates. Therefore considerations are done on database implementation.

3.3.1 Structured Databases – Microsoft Access

A typical structured, relational database system is good for handling information such as account numbers, account balances, names and addresses, times, dates, and other dollar amounts. The relational databases are good for this information because by providing a key, the database system can quickly query the database and retrieve the relevant information for a customer record. These relational systems are in which widespread use in financial and accounting applications and any application in which discrete, structured information is stored. A primary characteristic of this information is that typically the data fields are fixed length.

Examples of desktop relational databases systems include Microsoft Access, Borland dBase and Paradox, and Lotus Approach. Client/server examples include Oracle, Sybase, DB2 and similar systems. At the core of these systems is the concept of tables, consisting of rows and columns. Each column typically stores a certain type of the information such as name, address, or account balance. Each row of the database represents one record, grouping the fields from the different columns. As can be told, the structure of a relational database system is very rigid because the data model id based on rectangular, tabular data.

3.3.2 Unstructured Databases – Lotus Notes

A typical unstructured database is based more on textual information. This information does not lend itself well to fixed-length data fields. Consider a customer support application, in which the information stored is a description of the customer's problem or request. This description is narrative in nature and can actually be very long. This description could be stored so that many customer representatives could all access this common knowledge base.

Notes databases are different from typical relational databases with which might be familiar. The primary difference is that Notes databases are really designed to handle unstructured data, whereas traditional relational databases are optimized for structured data. This characterization is not absolute because traditional databases can handle unstructured data, and Notes can handle structured information

Notes database provides features that are optimized for the handling of textual information. The elements of a Notes database, in fact, are called documents and are the fundamental unit of storage within a Notes database. A Notes document is somewhat akin to a record in a traditional relational database, but because Notes documents can store this unstructured (as well as structured) information, they are very powerful. In addition, documents from one database can be related to other documents in the same database or other database.[9]

3.3.3 A comparison

Lotus Notes and relational database management systems (RDMS) implement each other because they are designed to support different application types. Understanding the difference helps to decide when an application requires Notes.[14]

RDMS	**Lotus Notes**
Based on a transaction-processing model.	Based on a document organizational model.
Use structured data.	Use semi-structured data elements (rich text, graphics, etc).
Offer real-time access to data.	Use periodic replication for updates.
Locate data using ad hoc queries.	Locate data using views (predefined queries).
Centralized within a single enterprise.	Share resources between enterprises.
Access via SQL.	Access via full-text search.

3.4 Functional Requirements

Functional requirements are a set of functions that are required to be included into the toolkit. Functional requirement describes an interaction between the toolkit and its environment. In some cases, the functional requirements may also explicitly state what the toolkit should not do.

3.4.1 Retrieve

The toolkit provides two components : graphics and templates. Users are allowed to copy the graphics or templates by click a COPY button.

3.4.2 Display

The toolkit provides two types of display : record listing and full graphics display. The record listing display is a display of the entire library in words with descriptions. Whereas the full graphics display shows the entire graphic in a full page.

3.4.3 Collect. Delete and Modify

User are allowed to add new graphics and templates to the toolkit's database. This module also enable users to delete and modify the existing graphics and templates in the database.

3.4.4 Favourite

The toolkit enables the users to keep their favourite graphics or templates into a folder. It is to allow the users can retrieve their favourite ones easily.

3.4.5 Help

Help provides help to assist users in using all the features mentioned above.

3.5 Non-functional Requirements

Non-functional requirements are the constraints under which a toolkit must operate and standards that must be met by the delivered toolkit. That is, a non-functional requirement or constraints describes a restriction on the toolkit that limits our choice for constructing a solution to the problem. These constraints usually narrow our selection of language, platform, or implementation techniques or tools.

3.5.1 User Friendly and Ease of Use

NGG provides a use-friendly interface, that is a graphical user interface for retrieving. It consists of buttons, which is very easy to use with a mouse click.

3.5.2 Reliability

The toolkit is reliable and is not cause unnecessary and unplanned downtime of the overall environment. A toolkit is said to have reliability if it does not produce dangerous or costly failures when it is used in a reasonable manner, that is, in a manner that a typical user expects is normal. This definition recognizes that a toolkit may not always be used in the ways that the designer expects.

3.5.3 Correctness

Correctness is the extent to which a program satisfies its specification and fulfills the customer's mission objectives.

3.5.4 Efficiency

Efficiency in a computer terminology means a procedure that can be called or accessed in an unlimited numbers of times to produce similar outcomes or output at creditable speed.

3.5.5 Simplicity

Simplicity refers to keep forms and screens properly uncluttered in a manner that focuses the user attention.

3.5.6 Reusability

The extent to which a program or parts of a program can be reused in other applications – related to the packaging and scope of the functions that the program performs.

3.5.7 Maintainability

A product is maintainable if the programs are easily to modify and test when updating to meet requirement, correcting errors, or move to a different computer system.

3.6 Current Flow without Graphical Toolkit

Figure 3.1 shows the system flow when using Lotus Notes to develop a Notes application without any other graphical toolkit. Developers can use the graphics or templates, which have been provided in Lotus Notes features. If developers want to use some other graphics, they have to import from other sources.

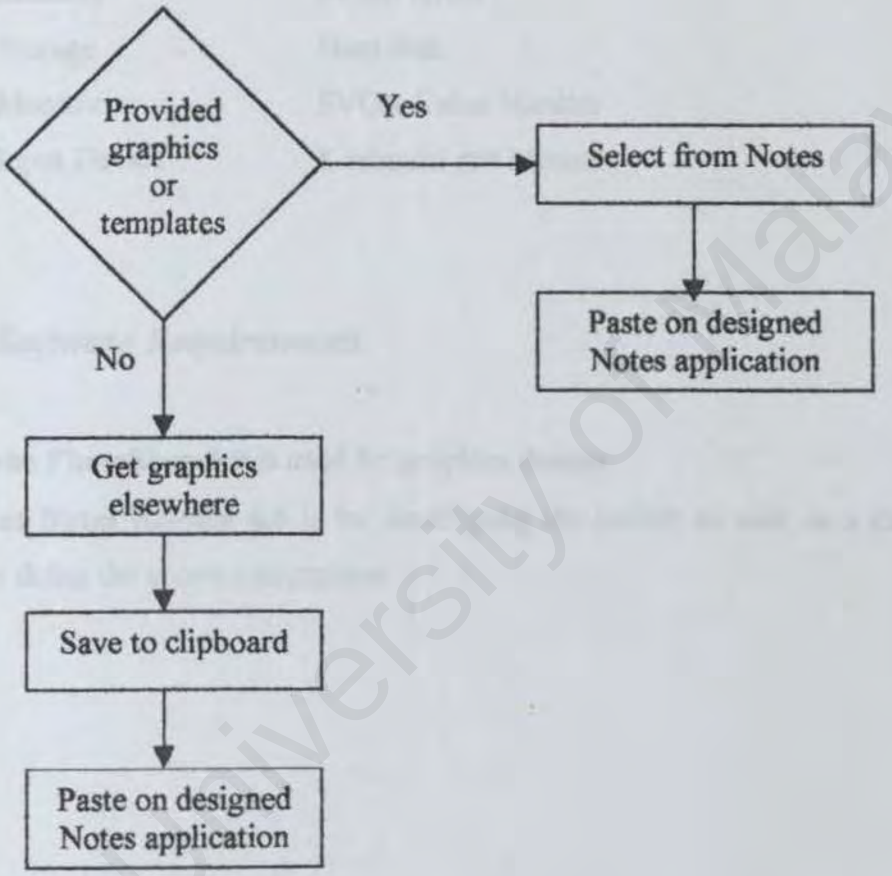

Provided graphics or templates
Yes
Select from Notes
Paste on designed Notes application
No
Get graphics elsewhere
Save to clipboard
Paste on designed Notes application

Figure 3.1

3.7 Run Time Requirements

3.7.1 Hardware Requirement

The following is the hardware requirements:

Processor (CPU)	: Intel Pentium 133 MHz
Memory	: 64 Mb RAM
Storage	: Hard disk
Monitor	: SVGA Color Monitor
Input Device	: Keyboard and Mouse

3.7.2 Software Requirements

- **Adobe PhotoShop 5.0** is used for graphics design.
- **Lotus Notes Release 4.6** is for developing the toolkit as well as a database after doing the above comparison.

CHAPTER 4

SYSTEM DESIGN

Chapter 4 SYSTEM DESIGN

Design is the creative process of the problem into a solution: the description of a solution is also called design. Modularity is a characteristic of a good design, the components have clearly defined inputs and outputs, and each component has a clearly stated purpose. Thus, to design a toolkit is to determine a set of components that satisfy a specified set of requirements.

In this chapter, the detail of how this toolkit will meet the requirements identified during requirement phase is described. The design of the NGG has considered the following issue:

- Toolkit data flow design
- Toolkit functionality design
- Database design
- Interface design
- Data entry procedures design

4.1 Overview of NGG Architecture

NGG is designed under an architecture as in Figure 4.1. Inside the NGG toolkit consists of two components and NGG itself is a database. For graphics component, NGG toolkit is related to other Notes databases with clipboard. Any selected graphics from NGG toolkit that need to be used on other Notes databases are using clipboard as a media to both databases. Wanted graphic from NGG is saved into clipboard and then pasted on the other database. For templates component, template database will have to be detached into Notes data directory (c:/notes/data/). Notes data directory is the media of both NGG toolkit and other Notes databases. Other Notes databases can inherit the design of the template database from NGG toolkit. However, both components in NGG are definitely independent.

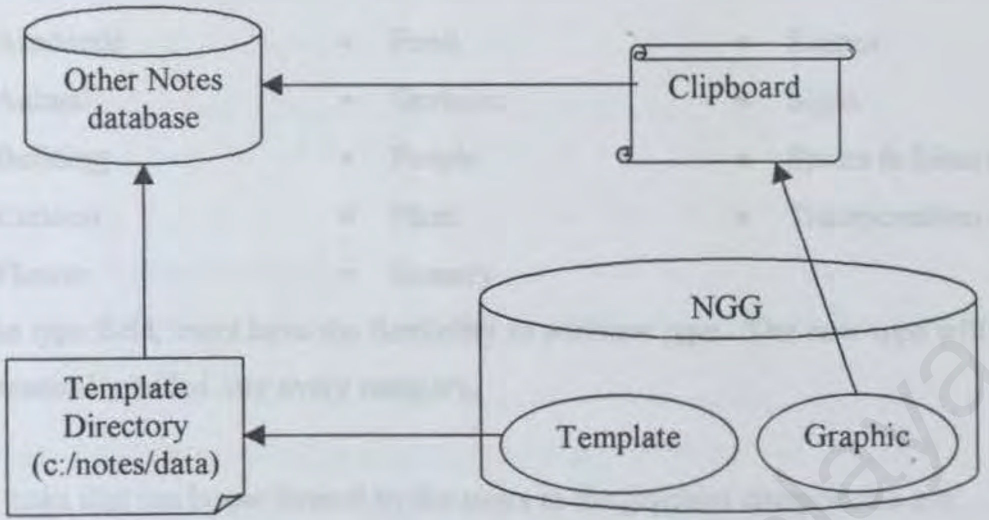


Figure 4.1 NGG Architecture

4.2 Overview of NGG Toolkit Components

Two components have been considered in the system analysis. There are graphics component and template component. Both components have some similar functions, but the contents of these two components are totally different and not depending to each other.

4.2.1 Graphics component

The graphics component's main page should consist of all the graphics categories and the users can select any of the categories. There are six fixed categories of graphics provided in this component:

- Background
- Button
- Frame
- Icon
- Line
- Picture

Besides that, users can always put the graphics, which are not in those categories in the "other" category.

For each category, there are divided into many types. Types that have been provided are:

- Academic
- Animal
- Building
- Cartoon
- Flower
- Food
- Gestures
- People
- Plant
- Scenery
- Shapes
- Signs
- Sports & leisure
- Transportation

In the type field, users have the flexibility to add new type. The new type will be automatically added into every category.

The tasks that can be performed by the users in the graphics components are:

- Add new graphics.
- View any of the selected graphics.
- Copy the selected graphics into clipboard for further usage.
- Edit the graphics document (e.g. name and description of the graphics), but not the graphics.
- Delete any of the selected graphics.
- Add favourite graphics into a folder.
- Get help on how to use this component.

4.2.2 Template component

Template component has less content than graphics component. There are only two categories, which are forms and navigators. Types of the categories are graphics and non-graphics. Forms and navigators are stored in this component as attachments. Therefore, form and navigator template can only be viewed by launching the attachment. The tasks that can be performed by the users in this component are slightly different from the component above. There are:

- Attach new form or navigator template into a document.
- Detach the selected template into a proper directory.
- Launch the selected template attachment to view the template.

- Edit the template document, but not the content of the template attachment.
- Delete any of the selected templates.
- Add favourite templates into a folder.
- Get help on how to use this component.
- Design form or navigator in the Notes design mode.

4.3 Toolkit Flow Design

The toolkit data flow design discussed here is from adding new graphics or templates to retrieve templates and copy graphics. There are two entities that are involved in the toolkit data flow: Notes developers and general users. General users are defined as those people without any knowledge on designing Notes Applications, but they might be the users of Notes Applications. NGG can be used for other purposes besides in designing Notes applications. Therefore one of the entities is general user. However the focus of NGG toolkit is still on Notes based applications. Thus, Notes developers who are the people developing the Notes Applications is treated as an entity. Figure 4.2 shows the context diagram of NGG toolkit.

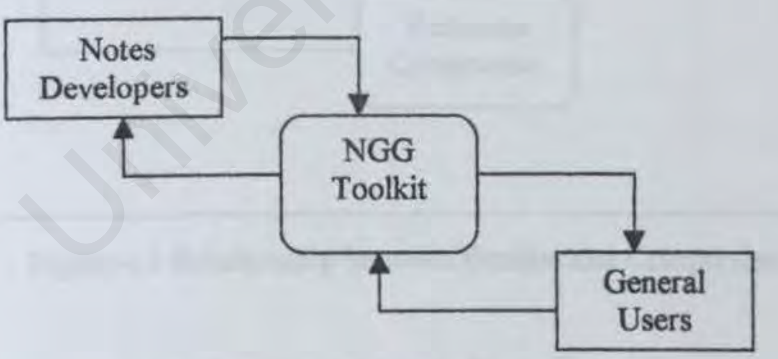


Figure 4.2 NGG Toolkit Context Diagram

As it has been mentioned above, NGG toolkit has two components (Figure 4.3 shows the relationship between entities and components). In the graphics components, general users and Notes developers are involved in performing some specific activities. These activities are divided into two parts, which are add new graphic and select graphics for further options. Every graphics is pasted on a form and saved as a document. Once the users save the new graphic, they are able to perform the select graphics activity. After select the wanted graphic, users will see the document of the graphic where the graphic is shown. Graphic can be copied to the clipboard by simply a click of "COPY" button. With a copy of graphic in the clipboard, users are able to paste it anywhere they prefer. In the edit mode of the selected graphic document, the document can be deleted, modified and added into a folder that keeps the users' favourite graphics. The flow of the graphics component can be viewed from Figure 4.4 and Figure 4.5.

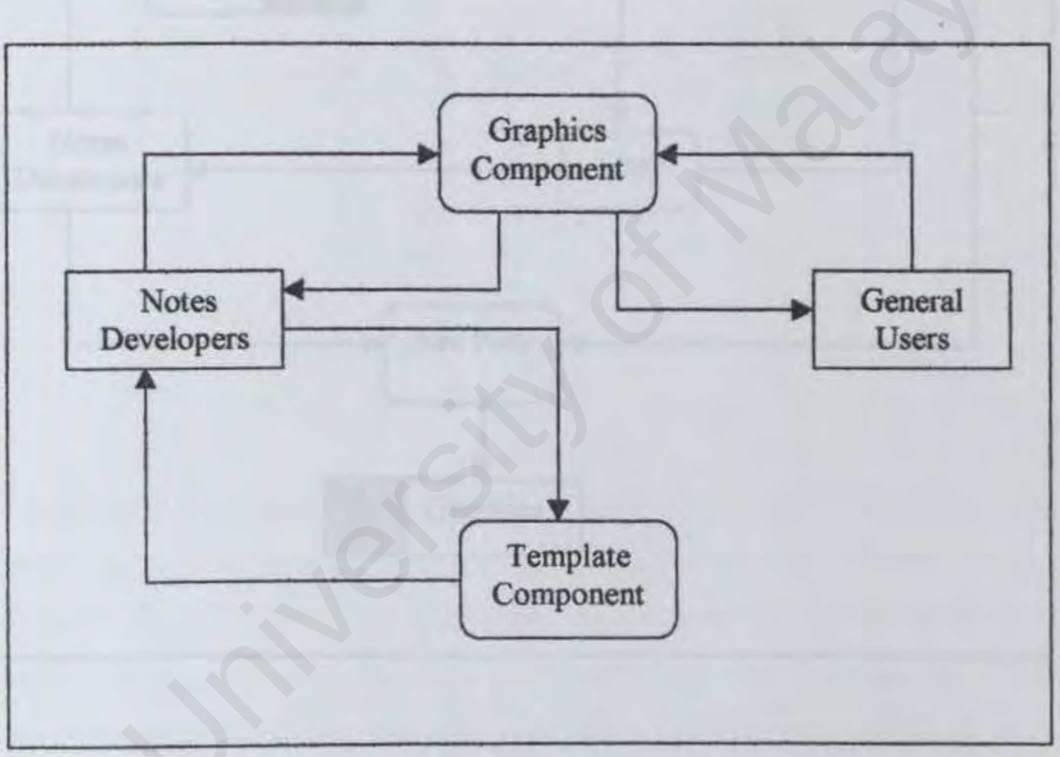


Figure 4.3 Relationship between Entities and Components

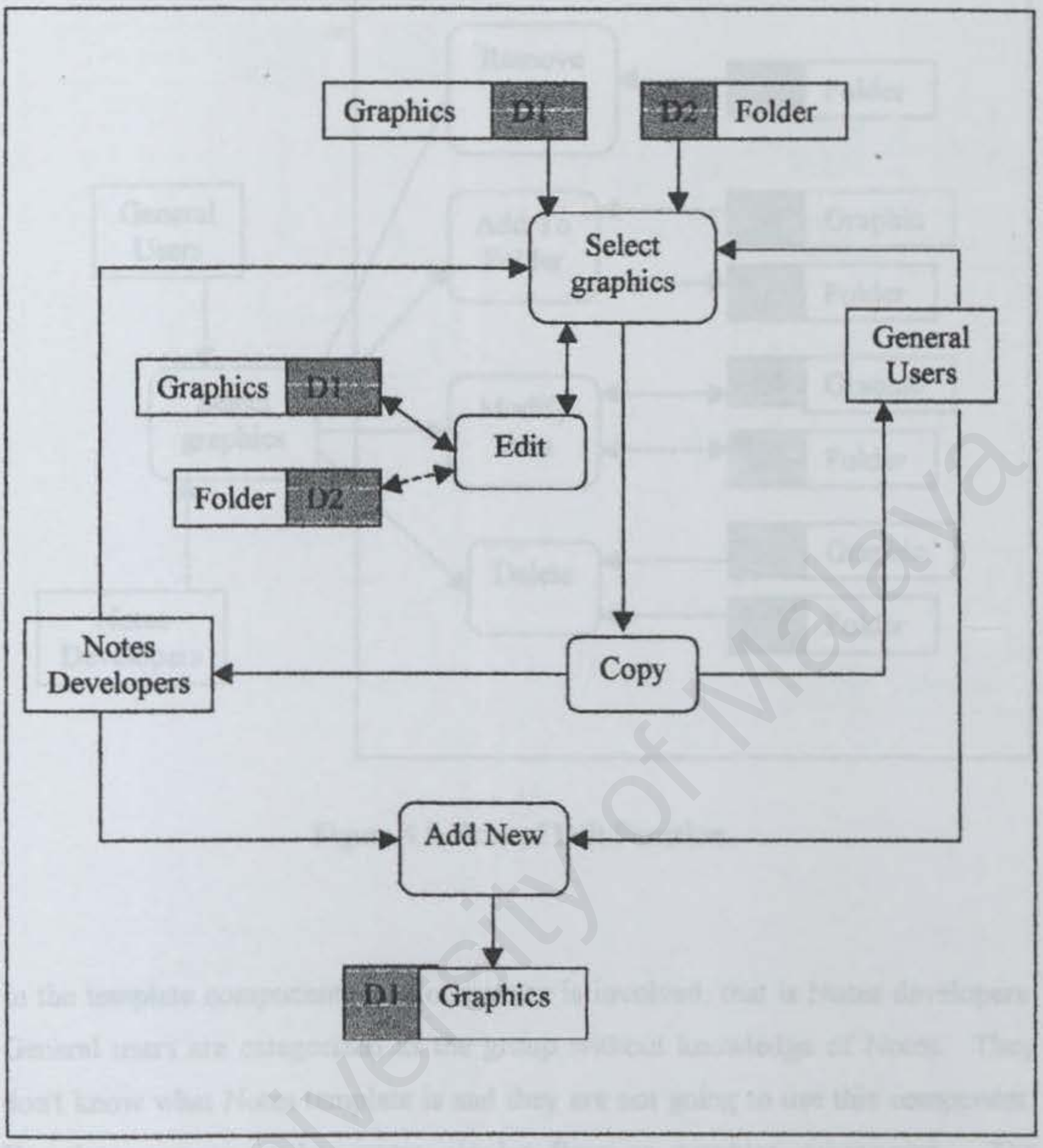


Figure 4.4 Flow of Graphics Component

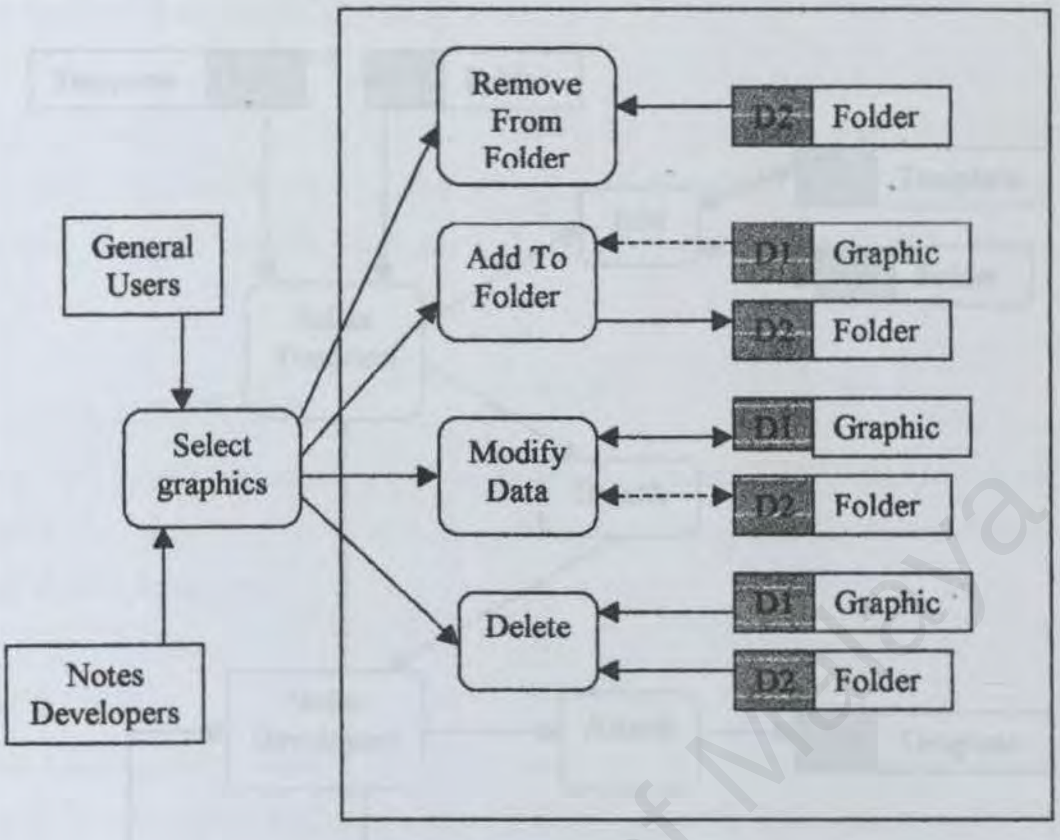


Figure 4.5 Flow of Edit Function

In the template component, only one entity is involved, that is Notes developers. General users are categorized as the group without knowledge of Notes. They don't know what Notes template is and they are not going to use this component. Template component has some similar flow as graphics component. The differences are in template component, template of form or navigator is saved as an attachment in a form. The attachment is actually a Notes template database file. The template can be viewed by detaching the file. The advantage of this component is users are allowed to switch to Notes design view. After switching to design view, Notes developers are able to design any forms and navigators in this toolkit. Notes developers can switch back to the template interface after their design work is done. Figure 4.6 illustrates the flow of template component.

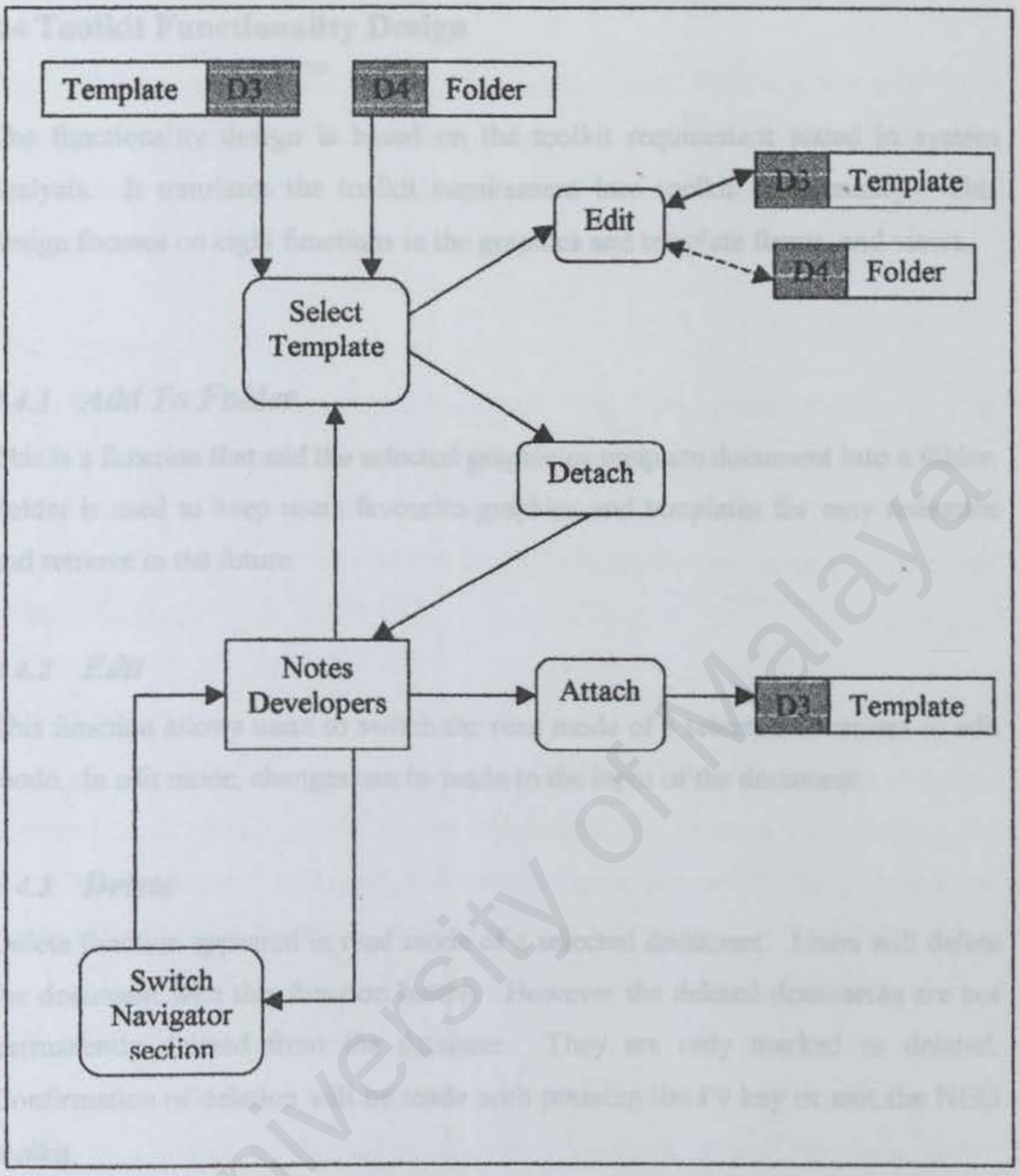


Figure 4.6 Flow of Template Component

4.4 Toolkit Functionality Design

The functionality design is based on the toolkit requirement stated in system analysis. It translates the toolkit requirement into toolkit functionality. This design focuses on eight functions in the graphics and template forms, and views.

4.4.1 Add To Folder

This is a function that add the selected graphic or template document into a folder. Folder is used to keep users favourite graphics and templates for easy reference and retrieve in the future.

4.4.2 Edit

This function allows users to switch the read mode of a selected document to edit mode. In edit mode, changes can be made to the input of the document.

4.4.3 Delete

Delete function appeared in read mode of a selected document. Users will delete the document with this function button. However the deleted documents are not permanently deleted from the database. They are only marked as deleted. Confirmation of deletion will be made with pressing the F9 key or exit the NGG toolkit.

4.4.4 Save

This is a function that use to save new graphic or template documents. Every changes that have been made during editing need to be saved with this function. This function will appeared in edit mode only.

4.4.5 Close

This is a function to close a selected document from edit mode or read mode without saving the document.

4.4.6 Copy

This is an important function in the graphics document. This function will copy the graphics that appeared in the document to the clipboard. After clicking the copy button, graphic that is copied can be used in anywhere besides Notes application.

4.4.7 Attach

Attach function is very important in template form. Whenever user creates new template, user has to attach the template database file into the attachment field. Attach function will prompt user to the create attachment window to select the file that want to be attached in the document. This function will appeared in the edit mode.

4.4.8 Detach

This function is important in the template document. Users retrieve the template database file with this function. Users is prompted to a window after clicking the detach button. Users are allowed to choose the space where they want to save the template database file to. Detachment is made in read mode.

4.4.9 Remove From Folder

To remove a document from a folder, this is the function should be used. This function is appeared on the action section in the graphics and templates folder views. This function will only remove the document from folder without deleting the document from the database.

4.4.10 Switch Design

This is a function that switches the interface of navigation section to Notes design view, and from design mode switch back to navigation section. This function can only be used in template component.

4.4.11 Add New

In graphic and template components, add new function is used to add new graphics and templates. This function will prompt to an empty form for users to fill up.

4.5 Database Design

Data storage is the most important element in this toolkit. The design of data storage must be able to achieve the objective of the data availability, data integrity, efficient data storage, efficient updating and retrieval and purposeful information retrieval. Data in this toolkit are store in Notes database. Table 4.1 shows the list of forms created for the two components.

Database's Title	Form	Functions Applicable	Fields
NGGalery	GraphicsF	• Add to Folder • Close • Copy • Delete • Edit • Save	• GName • GDescription • GCategory • GType • GImej
	TemplateF	• Add to Folder • Attach • Close • Delete • Detach • Edit • Save	• TName • TDescription • TCategory • TType • TImej

Table 4.1 Database Design

4.6 User Interface Design

The quality of system input determines the quality of system output. It is vital that input forms and screens be designed with this critical relationship in mind. Well designed input forms and screens should meet the objectives for effectiveness, accuracy, ease of use, consistency, simplicity, and attractiveness. All of these objectives are attainable though the use of basic design principle, knowledge of what is needed as input for the system, and an understanding of how the user should respond to different elements in the forms and screens.

4.6.1 Main page design

The main page display is designed to access two components, graphics component and template component. Help, about NGG and list of popular clipart web-site that can be accessed from this page. Figure 4.7 illustrates the main menu of NGG.

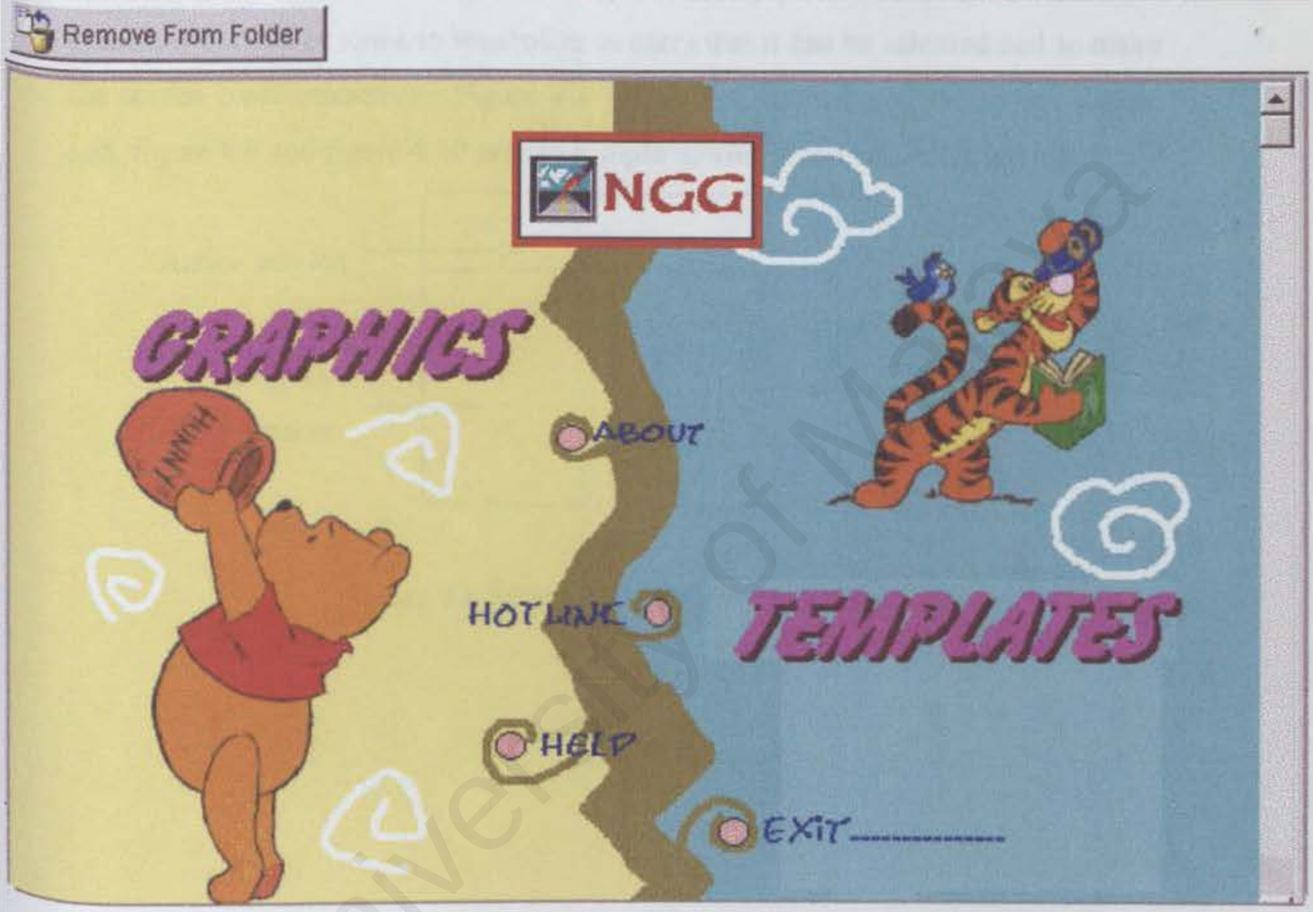


Figure 4.7 Main Menu

4.6.2 Component screens design

The screen display is simplified by dividing it to three sections. The right column of the screen contains the name of the graphics or templates (view section). The left column of the screen, which is the navigation section, contains available categories and major functions. Actions, on the top column are represented with graphical buttons or icons to symbolize to users that it can be selected and to make the screen more attractive. Figure 4.8 shows the structure of the screen design and, figure 4.9 and figure 4.10 are the sample screen of the two components.

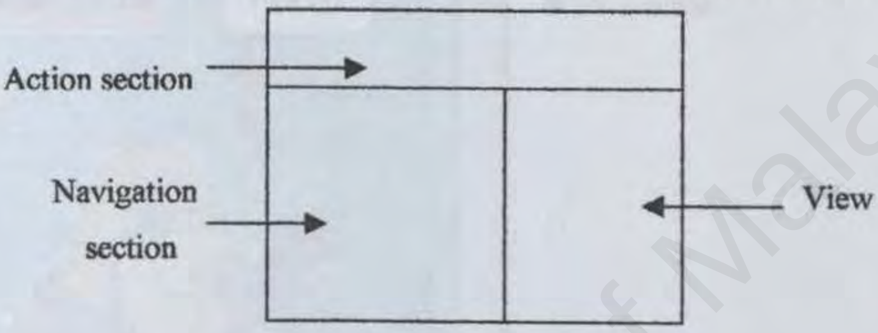
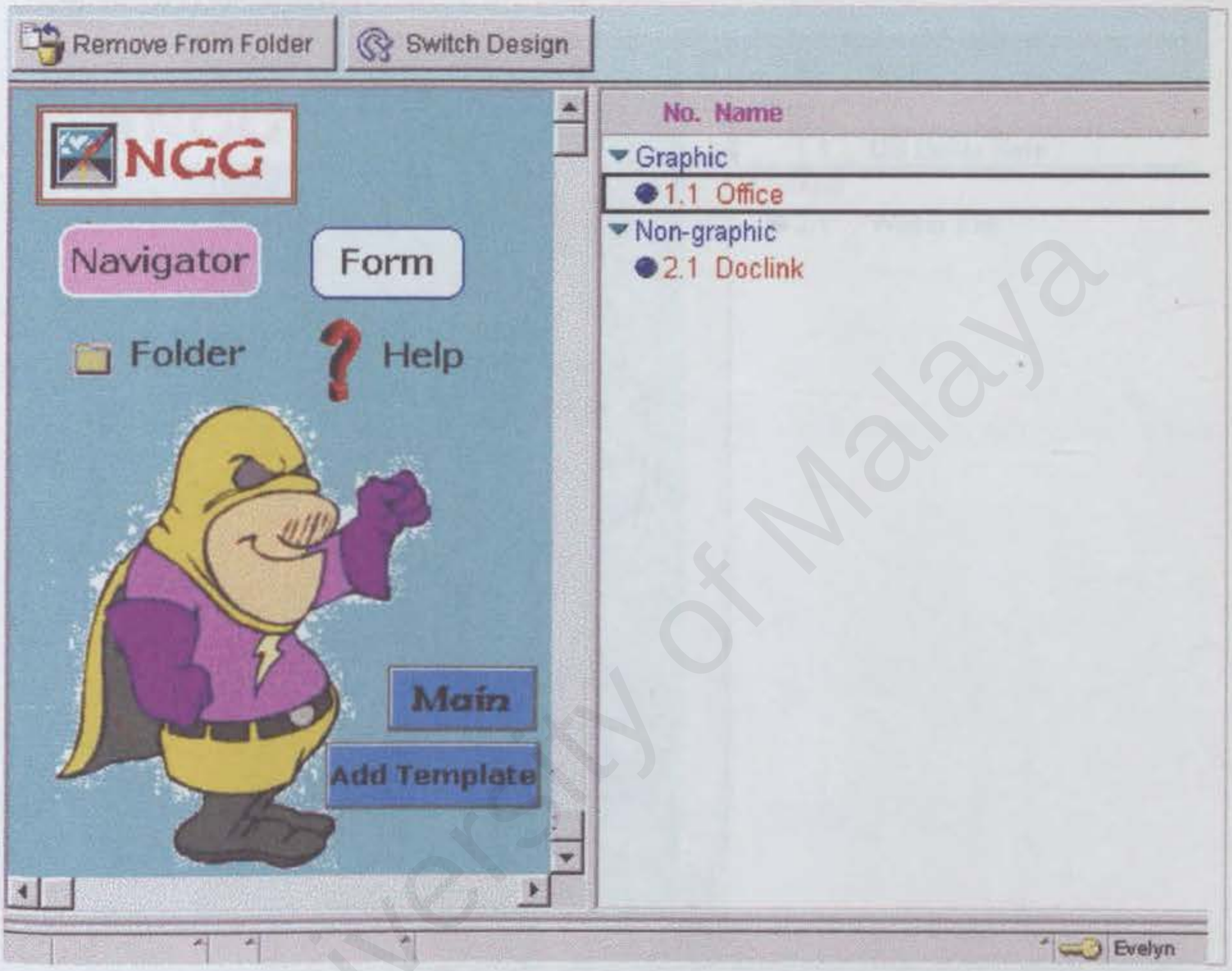


Figure 4.8 Structure of the screen design



No.	Name
	Graphic
1.1	Office
	Non-graphic
2.1	Doclink

Figure 4.9 Template Menu

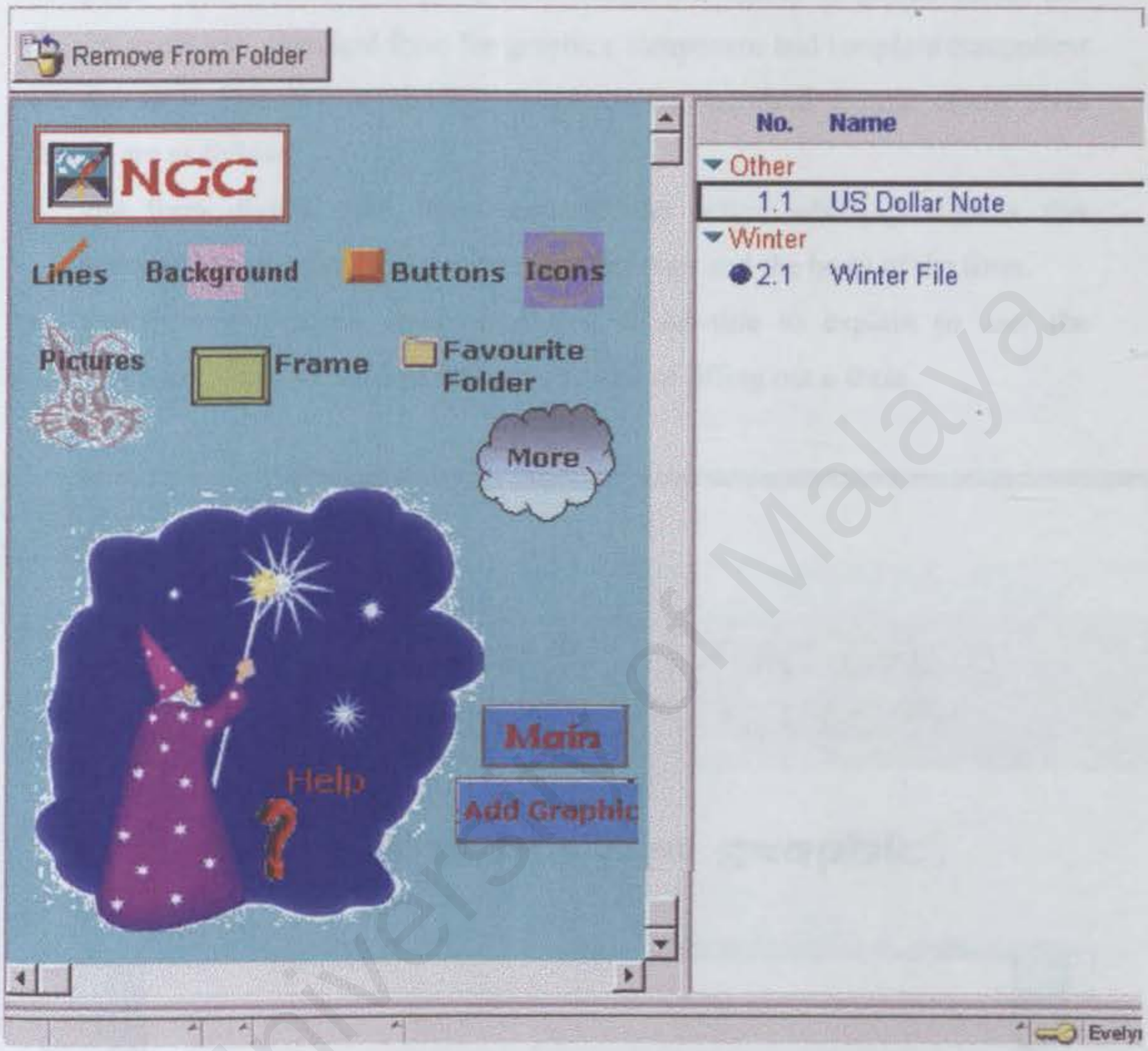
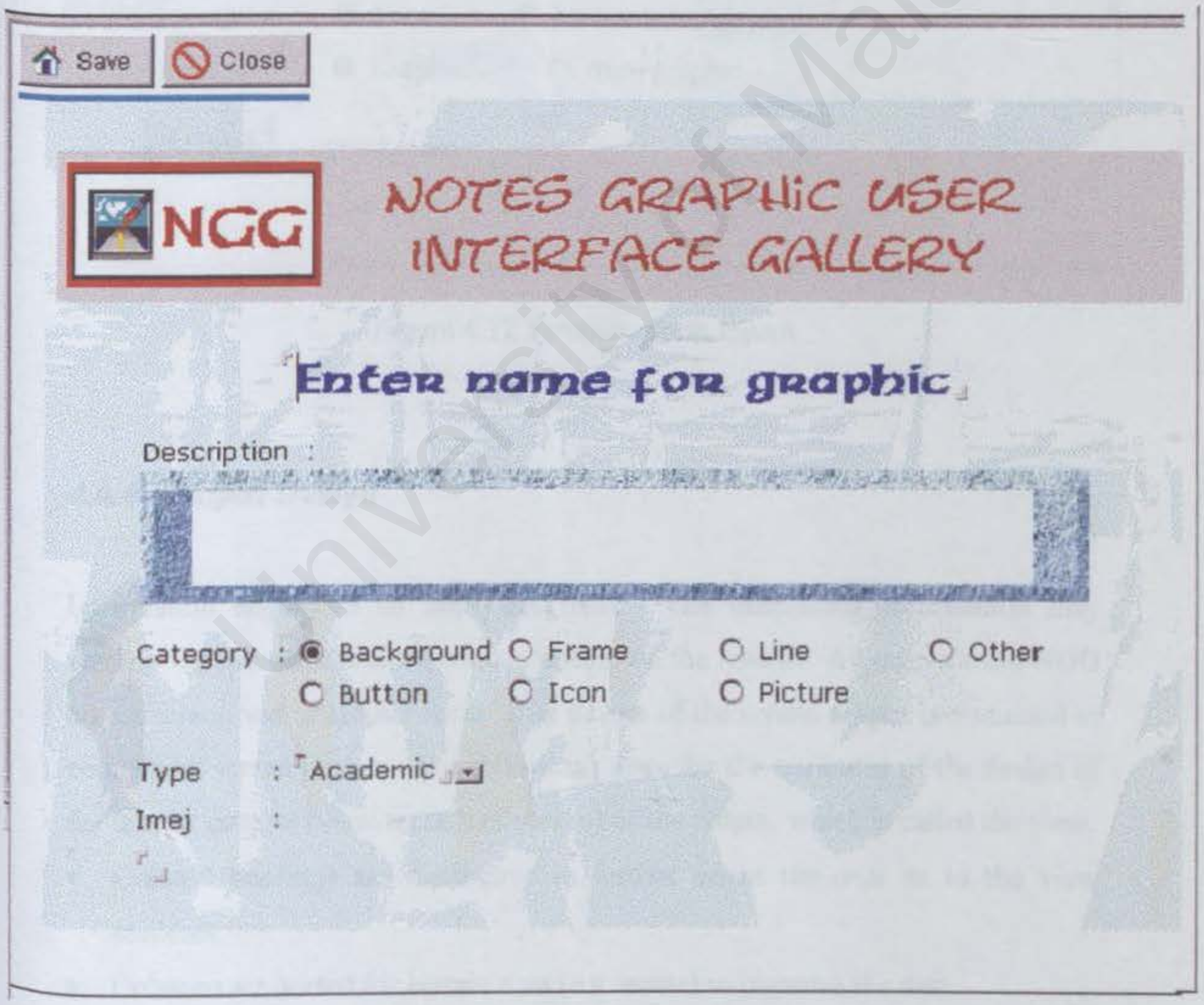


Figure 4.10 Graphics Menu

4.6.3 Form Design

Forms are use to capture information required that often will be input to the computer. The objectives of good form design are similar to screen design and also its accuracy. Standard form for graphics component and template component are shown in Figure 4.11 and Figure 4.12. The standard criteria of the form design are as follows:

- The form divided into three sections: the action which provide a few functions, the heading which give the NGG logo and the body of the form.
- The form entries are categorized with a sub-title to explain to user the relevance of the entries and make easy work of filling out a form.



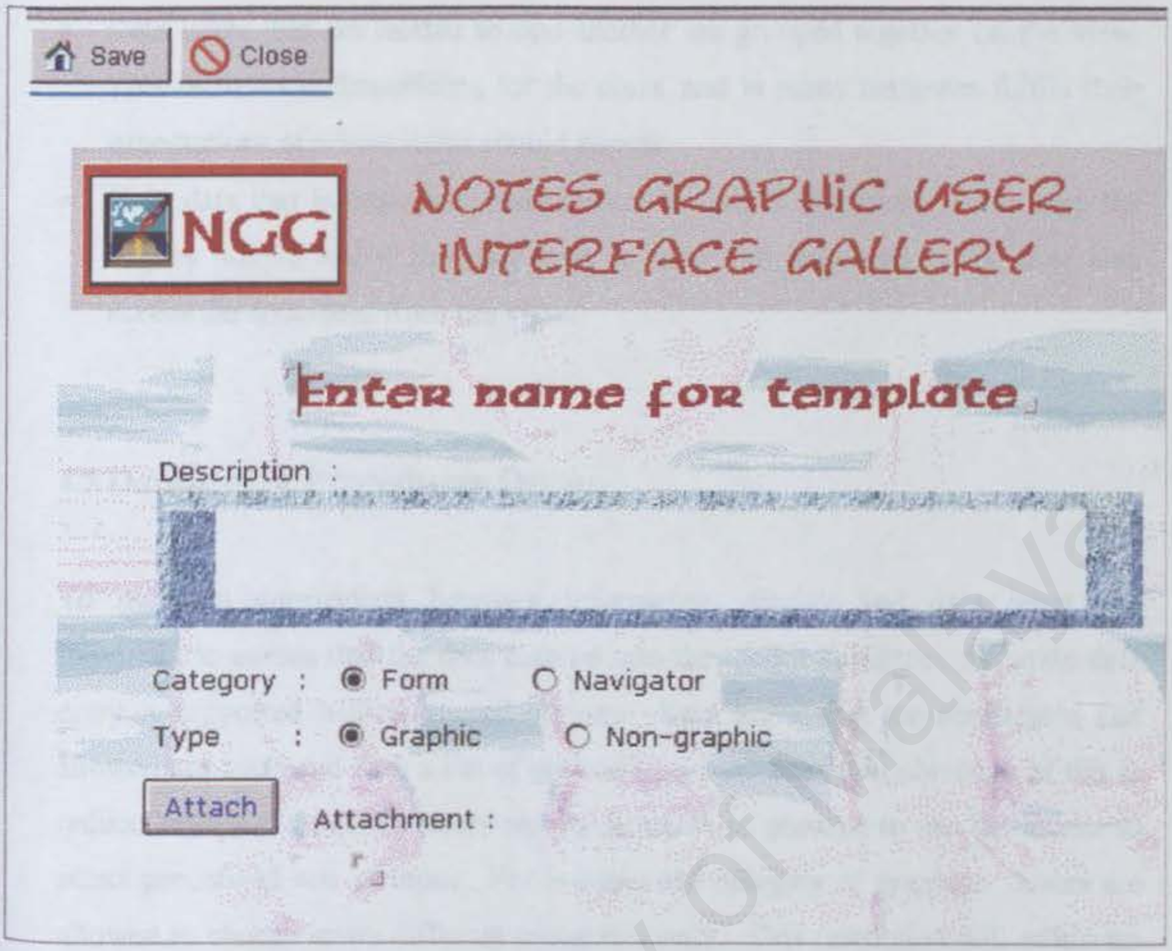
Description :

Category : ● Background ○ Frame ○ Line ○ Other
○ Button ○ Icon ○ Picture

Type : Academic

Imej :

Figure 4.11 Graphic form layout



NOTES GRAPHIC USER INTERFACE GALLERY

Enter name for template

Description :

Category : ● Form ○ Navigator

Type : ● Graphic ○ Non-graphic

Attachment :

Figure 4.12 Template form layout

4.6.4 Output Design

Information display is an important issue. The misplacing information may confuse the users and lead to misconception on the results. All output from NGG are on screen and in tabular form. The design of the screen output is discussed in component screen design. The followings describe the attributes of the design of the tabular content (view section of screen) of the output, which is called the view.

- Column headings are distinctive to further orient the user as to the view contents.
- Columns are sorted for certain data (e.g. name) to organise the data.

- Data items that are related to one another are grouped together on the view. This facilities understanding for the users, and in many instances fulfils their expectations of where items should appear.
- Only data that is important and relevant to user is displayed. This keep the display simple and if the user wish to view full information, the user may access the document from the view.

4.7 Data-Entry Procedures Design

To maintain consistency between information display and data input, it's important to ensure that the data entered into the toolkit accurate. Accurate data entry is supported with restricted options. Data items that are predictable and allowed are restricted with a list of options to select. The main objective of this is reduce keyboard typing by users and try as much as possible to use the mouse to select predefined sets of input. For instance the category of graphics. Users are allowed to choose seven different categories only. This restriction will minimize the number of input actions required of the users. The list entries are sorted to allow ease of search of item to select. The validity of the selection items is insured.

4.8 Help Page

There are two help pages in the NGG toolkit. One help page is to explain the way to use graphics component and another one is for template component. Help page can be access from each component screen in the navigation section.

CHAPTER 5

IMPLEMENTATION & TESTING

CHAPTER 5 IMPLEMENTATION AND TESTING

The implementation process comprises the translation of the system design structure to a computer-readable system. It is the developer's responsibility to evolve the particular system from scratch design to a runable application. The major tasks in this phase includes coding, testing (test the program) and debugging (fix the bug occurred in the program).

5.1 Development of GUI

Navigator is used in all the user interfaces in the NGG toolkit. To make the appearance of these pages look more attractive; buttons and pictures are being used. By clicking the button, it will call a function to do the task of changing the content of view section or point to another navigator.

5.2 Development of Creating Database

The development of NGG toolkit is initiated at the creation of new Notes database using the Notes Designer. The most ambitious way to create a database is to start from scratch. Creation of database begins with a blank file that has the extension .NSF. This file contains no forms, fields, views, or other elements. All of the forms, fields, views, and other elements that needed for NGG are created. The next few sections explain how the content and structure of database are created and customized.

5.3 Development of Creating Forms

NGG forms are created using Notes Designer. A form is a framework for entering and viewing information in a database. A Notes database contains documents created from one or more forms. When a form is designed, components to show and how they should look to users are decided upon. Input forms are created with Notes application development resources such as computed field and input validation formulas.

5.4 Development of Creating Views

Standard-style views display document summarizes in rows and columns for users to find or see the documents they want to read. To open a document for reading, users open a view represented by one row. Views are created for every category of graphics and templates. All the views are created with forms associated to it.

5.5 Coding

5.5.1 LotusScript

Coding is a process that translates a system design into a programming language realization, which is a machine-readable form. LotusScript is an embedded language with object-oriented extensions. For more interactive actions, LotusScript is written to trigger some actions. Example of LotusScript being used in NGG is as the following. It's used to copy the graphic in a rich text field to the clipboard.

```
Sub Click(Source As Button)
    Dim workspace As New NotesUIWorkspace
    Dim uidoc As NotesUIDocument
    Set uidoc = workspace.CurrentDocument
```

```
            UIDoc.editmode =  "True"
            Call uidoc.GotoField( "Imej" )
            Call uidoc.SelectAll
            Call uidoc.Copy
            UIDoc.editmode =  "False"
        End Sub
```

5.5.2 Formula

Formulas are expressions that have program-like attributes. Formulas are used in certain toolkit functionality. Document selection is required in NGG when selecting document to be displayed in a view. The Notes Formula keyword SELECT is used in this selection formula. The SELECT keyword defines criteria for the selection of documents that runs a formula, in a view. Table 5.1 shows the formula used in every function.

Functions	Formulas
Attach	• @If • @AttachmentNames • @Do • @Command([EditGotoField];"TImej") • @Command([EditInsertFileAttachment]) • @Command([ViewRefreshFields]) • @Prompt([OK];"Error";"Only one file attachment is allowed per document.")
Switch Design	• NavName:=@Trim (@Name([CN];@Subset(@DbName;1))+@Subset(@DbName;-1)+"NavState"); • @If • @Environment (NavName)="0":""; • @Do • @SetEnvironment (NavName;"1");

	• @Command([OpenNavigator];"TemplateN";"0")) • @Command([ViewNavigatorsFolders])) • @Command([RefreshHideFormulas])
Detach	• @Command([AttachmentDetachAll])
Delete	• @Command([EditClear]) • @Command([FileCloseWindow])
Save	• @Command([FileSave]) • @Command([FileCloseWindow]) • @PostedCommand([Compose]; "TemplateF")
Close	• @Command([FileCloseWindow])
Access About NGG	• @DialogBox("AboutNgg"; [AutoHorzFit] : [AutoVertFit] : [NoCancel] : [NoNewFields]: [ReadOnly])
Add New	• @Command([Compose]; "TemplateF")

5.5.3 Simple actions

Simple actions are predefined Notes functions that can be strung together to carry out a desired task. The following are the simple actions that have been used in developing NGG :

- Edit Document
- Move To Folder
- Remove From Folder
- Open a View
- Open another Navigator
- Alias a Folder
- Open URL

5.6 Toolkit Testing

The objectives of testing are:

- To ensure the toolkit is developed according to its specification that is descriptions of user's requirement and expectations.
- To detect all the errors that might reside in the toolkit and consequently remove all the detected errors.

Generally, two types of testing are being used in this project. They are component testing and toolkit testing.

5.6.1 Component testing

Individual components are tested to ensure that they operate correctly. Because graphics component and templates component are two independent component, they can be tested as a stand-alone entity. The level which is tested is field level where there is a rich text field to contain graphics or attachments. The validation of the input in this field is tested. View selection formula is tested as well to ensure that the documents displayed are the same as the expected documents to be displayed.

5.6.2 Toolkit testing

The testing process is concerned validating that the toolkit meets its functional and non-functional requirements. This is the last procedure to ensure that the whole toolkit is able to function properly. In this testing procedure, general users and Notes developers are invited to use the toolkit as to test the toolkit. They have tried to use NGG toolkit for several times. After using NGG toolkit, they have given some comments and recommendations for changes and improvements. Most of the recommendations were focus on user friendliness. All the opinions given by the users are considered and some of the new requirements have been met up after the testing.

CHAPTER 6

EVALUATION & CONCLUSION

CHAPTER 6 EVALUATION AND CONCLUSION

During the development of NGG, several problems were encountered. Some of them were ultimately solved, and some will be solved in due time. The following section explains system's strengths and limitations as well as the problems and solutions to overcome them. It includes suggestions for improving the present system and for potential enhancements.

6.1Toolkit strength

6.1.1 User friendly

NGG provides an attractive and ease of use interface for users. The set of standard GUI control objects, such as icons, buttons, test box, radio button, dialog list, and Window-oriented point and click interface, have been applied to give users fast access. Therefore, the time for users to learn and use the toolkit can be reduced.

6.1.2 Accessibility and ease of use

NGG provides both the general users and Notes Developers fast access since the layouts and commands are easy to use, simple to learn and understandable. Moreover, NGG is easily manipulated using either the keyboard or mouse.

6.1.3 Better problem resolution

NGG minimize the time spent in searching for graphics and designing user interfaces as many graphics and templates are provided in NGG. Many problems come from designing forms and navigators, NGG eliminates this situation because templates are provided. Notes developers just have to inherit the design from the template database.

6.1.4 Portability

NGG can be installed in any computer as long as the computer has Lotus Notes application. NGG can be used by any people even they have no knowledge on Lotus Notes.

6.2Toolkit Limitations

Owing to time constraints, NGG does not include all the functions that might be provided by a graphical toolkit. Some functions have been simplified, truncated or pared.

6.2.1 No overall graphics view

In NGG toolkit, every graphics can only be seen by accessing the document that stores the graphics. There is no way to view all the graphics in one page.

6.2.2 Template database attachment limitation

Currently, to add a new template document in NGG, Notes developers have to create a new template database in Notes application before attaching to the NGG template form.

6.3Problems and Solutions

Throughout the process of developing NGG, there were a few problems encountered. The following explains the problems and solutions to overcome them.

- Finding graphics and creating template databases for the input are not easy works. Graphics might not be too small or too big. Sometimes graphics would paste well in the documents, but not in other application after being copied from the document.

- Navigator and form templates are not easy to be designed so that they looked like more dynamics when viewing them.
- Template databases are created in Notes application. All these databases will be saved at c:/notes/data directory. When creating new attachment in NGG, users have to find the root to get the template database file.
- There are many constraints in working LotusScript, @commands and formulas. For some objects, both scripts and formulas can be used; for others, only one or the other can be used. Either one of them is been used to avoid problems occurred.

6.4Future Enhancements

NGG should be improved, corrected and further enhanced to include more features and functionality in the future. The following are some of the recommendations to enhance to the current version.

6.4.1 Better view for graphics

In the future version, it is important if the toolkit could view all the same category graphics in one page instead of accessing every document to view a single graphic. This could make the work of choosing the prefer graphics easier and efficient. Future version of NGG should be able to compress the graphics into a thumbnail size and put in a page for better viewing.

6.4.2 Better development for template component

One of the limitations of the toolkit is troublesome in creating new template database for the input in NGG. As a future enhancement, the form and navigator templates can be created in NGG and copied to another database.

6.4.3 Latest version of development tools

In this project, development tool that used is Lotus Notes 4.6 whereas Lotus Notes R5 has been launched almost a year ago. The toolkit should be upgraded to latest version of Lotus Notes in future. This could let the new R5 users make use of this toolkit.

6.4.4 Working with other graphical software tools

For future enhancement, other graphical software tools, such as Adobe Photoshop, Paintbrush and Serif DrawPlus should be embedded in NGG as one of the feature. By doing this will allow NGG users to create and design graphics in NGG using these graphical software tools.

6.5Knowledge Gained

As this project is developed to provide graphics for user interface design, graphical software tools are used, such as Microsoft Paintbrush and Adobe Photoshop. Concepts on how to design interactive user interface are studied. Graphics cover a very wide range of contents. Therefore how to categorized graphics into smaller categories need to be learnt.

6.6Conclusion

NGG toolkit has been successful in attaining its objective of enhancing the designing Notes application user interfaces process. NGG was found to be user-friendly, easily understood, and effective. It simplifies the work of creating and designing graphical user interfaces not only in Notes application but also other applications. NGG can be installed and then used in any computer as long as there is Lotus Notes 4.6.

However, some minor limitations still exist in the toolkit. Future enhancements will eliminate the limitations and ensure the toolkit will become more powerful and useful. Other future features will add to the NGG's utility.

Throughout this project, a lot of experiences have been gained, new knowledge has been acquired and, more importantly, there has been an improvement in project management and communication skills. Having up-to-date knowledge and information becomes important in keeping up with the fast and ever changing fields in the information technology edge.

[11] Tropp Technology, 1998
http://www.progtips.com/

[12] Whitten, Bentley, Barlow, *System Analysis And Design Methods, International Student Edition, Third Edition*, Irwin.

[13] Pressman, R.S., *Software Engineering: A Practitioner's Approach, New York*: McGraw-Hill, Inc., 1992.

[14] Lotus Education, *Lotus Domino 4.5 and Notes 4.5 System Administration 1*, December 1997

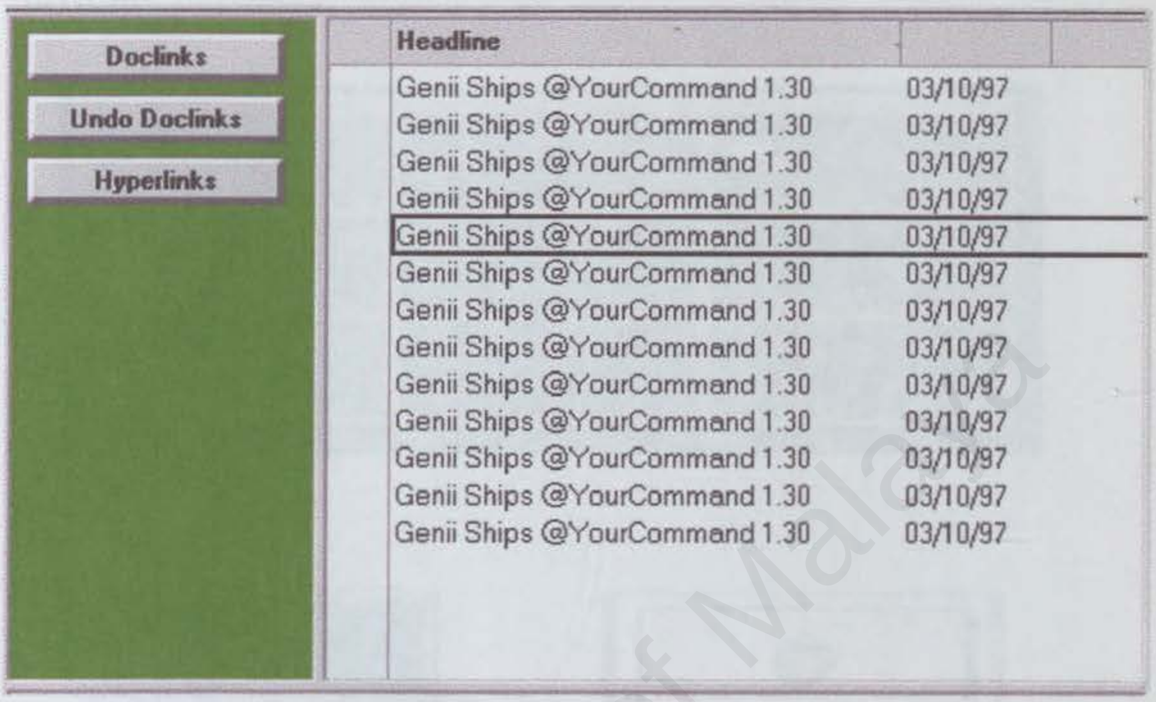
APPENDIX

Appendix A List of Questions

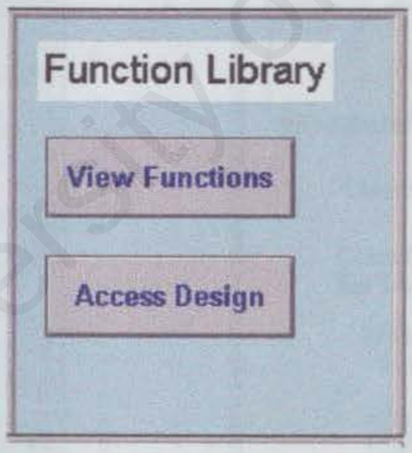
1. How do you develop user interface for Notes application?
2. How long do you take to complete all the user interfaces in an application?
3. What are the differences of views, forms, navigators, layout regions and subforms?
4. What are the features Lotus Notes provide for designing user interface?
5. What type of user interfaces normally you develop?
6. What are the differences between graphical and none graphical user interface?
7. What type of graphics you normally will include in the user interface?
8. What type of user interfaces normally preferred by the users?
9. Does Lotus Notes provide any graphics for its user interface design elements?
10. Where do you get the graphics if you need them for your user interface design?
11. Does Lotus Notes provide any features to import graphics from other software?
12. What are the difficulties and problems facing when you design user interfaces?
13. Have you use any of the graphics toolkit when developing the user interface?
14. Does the graphics toolkit help in designing user interface?
15. What kind of graphics the toolkit provide?
16. What types of functions are included in the toolkit?
17. How does the toolkit help you in designing the user interface?
18. Do you have any difficulties when using the toolkit?
19. What are the features you expect from a toolkit?

Appendix B Plain User Interfaces

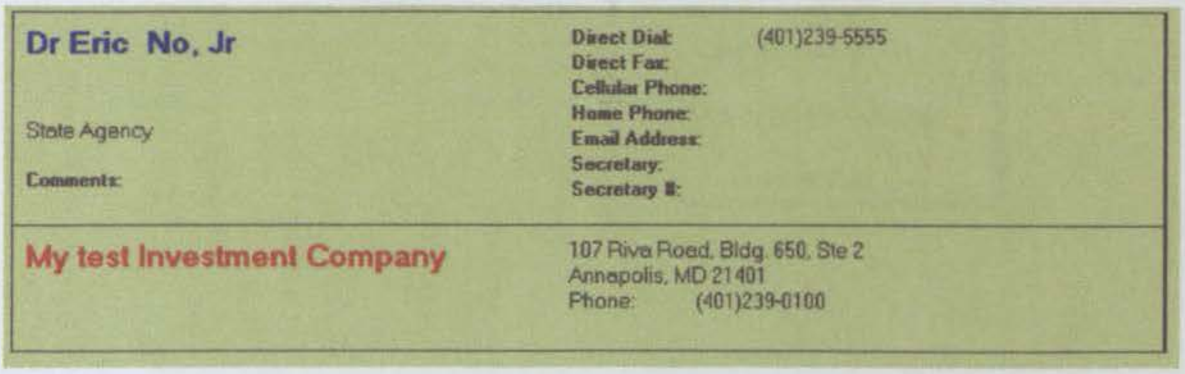
Navigator Interfaces



Headline	
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97
Genii Ships @YourCommand 1.30	03/10/97



Form Interface



Dr Eric No, Jr

State Agency

Comments:

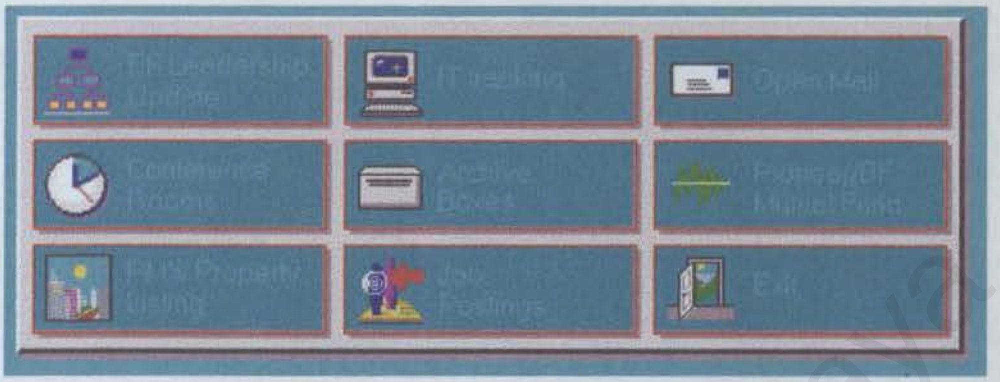
Direct Dial: (401)239-5555
Direct Fax:
Cellular Phone:
Home Phone:
Email Address:
Secretary:
Secretary #:

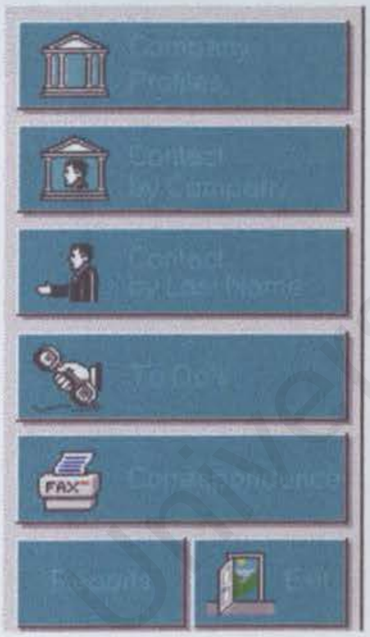
My test Investment Company

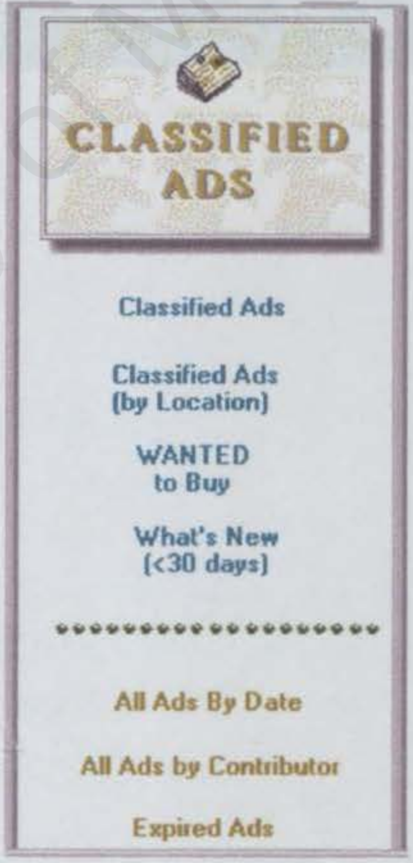
107 Riva Road, Bldg. 650, Ste 2
Annapolis, MD 21401
Phone: (401)239-0100

Appendix C Graphics User Interfaces

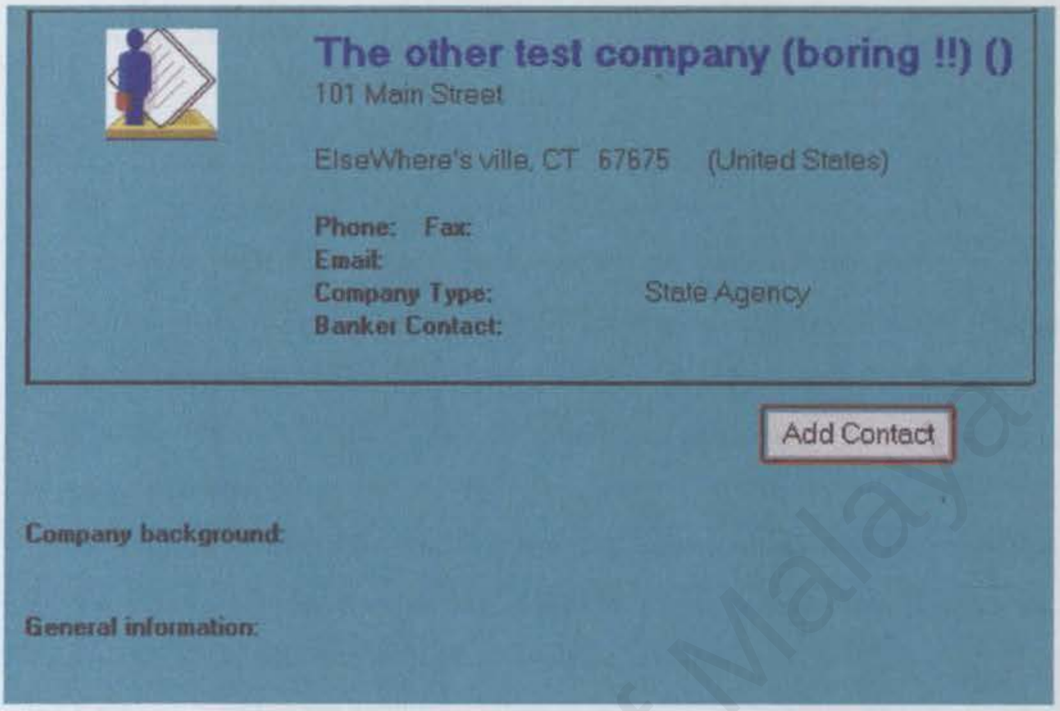
Navigator Interfaces







Form Interface



The other test company (boring !!) ()
101 Main Street

ElseWhere's ville, CT 67675 (United States)

Phone: Fax:
Email:
Company Type: State Agency
Banker Contact:

Company background:

General information:

Appendix D User's Guide

1. Getting Started

- An *about database* interface will be popped up after user activated NGG toolkit.
- User needs to press **ESC** button to access the *main menu* of the toolkit.
- In the *main menu*, there are two hotspot rectangles to access *graphics* menu and *templates* menu. User may **click** on the **hotspots** to access the menus.
- User may **click** on the other four red bullets that appeared in the middle page of the main menu. They are:
  - ➢ **About** : Display *about dialog box* that contains toolkit information.
  - ➢ **Hotlink** : Display a list of *URL Link* that provide free access of graphics.
  - ➢ **Help** : Display a *help guide* for user in using NGG.
  - ➢ **Exit** : *Quit* the toolkit.

2. Graphics Menu

- User may **click** on:
  - ➢ **NGG logo** - to access *about NGG* dialog box.
  - ➢ **Help** hotspot - to access *help guide* in using NGG.
  - ➢ **Main** button - to get back to *main menu* of NGG.
  - ➢ **Add Graphic** button - to access to add new graphics form.
  - ➢ 8 other **hotspots** that represent the categories of **graphics (lines, backgrounds, buttons, icons, picture, frame and more**) and favourite folder - to get the list of documents in the view section.

- To *access each document*, user may **double-click** on the selected document on the view section.

2.1 Get graphics from selected document

- Document will be opened as a form after user select the document from the view section.
- Click on the Copy button. The graphics is then copied to the clipboard and user may paste it at other Notes application and non-Notes application, such as PaintBrush.

2.2 Create new document

- User may click on the Add Graphics button from the graphics menu to access to a new graphics form.
- User may fill up all the fields (name, description, category and type) in the form.
- User needs to choose choices from the radio buttons that have been given in the form for category field.
- For type field, user needs to click on the Entry Helper button to see the choices in the Select Keywords dialog box, or press the key for the first letter of the type user wants to fill up the field. User is able to add new type in the Select Keywords dialog box. The new type will appeared in the dialog box in future.
- To paste a graphic on the Imej field, the graphic must first saved into the clipboard in other application.
- Place the cursor in the Imej field, right-hand click the mouse to choose paste or choose paste from the edit menu. The graphic will paste in the field if there is a graphic copied into the clipboard.
- Click Save button to save the document.
- User is able to continue adding new graphic as a new form is been popped out after saving a document.
- To exit from the graphic form without saving the form, click Close button.

2.3 Edit selected document

- After selecting a document from the view graphics section, document that appeared is in read mode.
- Click Edit button to change the document to the edit mode.

- User may change any information in the document.
- User can save the changes that are made when editing the document by clicking the Save button.
- User may exit the document by clicking the Close button.

2.4 Add selected document into a folder

- User may add his or her favourite graphic into a folder by clicking the selected document and dragging it to the folder in the graphics menu.
- Another way of performing the task is open the document and click on the Add To Folder button.

2.5 Remove selected document from a folder

- User may remove the selected document from a folder. User only has to click on the Remove From Folder button on the top of the graphic menu.

2.6 Delete selected document

- There is two ways to delete a document. The easiest way to delete the document is to select the document in the graphics view section, press Delete.
- Another way is open the document and click on the Delete button.
- To confirm the deletion of the document. Press F9 and then click on OK button to confirm. Anyway, confirmation of deletion will pop up before exiting the toolkit.

3. Template Menu

- User may **click** on:
  - ➢ **NGG logo** - to access *about NGG* dialog box.
  - ➢ **Help** hotspot - to access *help guide* in using NGG.
  - ➢ **Main** button - to get back to *main menu* of NGG.
  - ➢ **Add Template** button - to access to add new template form.

- Two **hotspots** that represent form template and navigator template, and favourite folder - to get the list of documents in the view section.

- To *access each document*, user may **double-click** on the selected document on the view section.

3.1 Detach template from selected document

- Document will be opened as a form after user select the document from the view section.
- Click on the Detach button. User is prompted to a save attachment window. User may choose the directory where user wants the file to be saved at.

3.2 Create new document

- User may click on the Add Template button from the template menu to access to a new template form.
- User may fill up all the fields (name, description, category and type) in the form.
- User needs to choose choices from the radio buttons that have been given in the form for category field and type field.
- To attach the template database file into the document, user may click on the attach button. User is prompted to create attachment window where user can choose the file wanted to be attached.
- Click Save button to save the document.
- User is able to continue adding new template attachment as a new form is been opened after saving the document.
- To exit from the template form without saving the form, click Close button.

3.3 Edit selected document

- After selecting a document from the view template section, document that appeared is in read mode.
- Click Edit button to change the document to the edit mode.
- User may change any information in the document.

- User can save the changes that are made when editing the document by clicking the Save button.
- User may exit the document by clicking the Close button.

3.4 Add selected document into a folder

- User may add his or her favourite template into a folder by clicking the selected document and dragging it to the folder in the template menu.
- Another way of performing the task is open the document and click on the Add To Folder button.

3.5 Remove selected document from a folder

- User may remove the selected document from a folder. User only has to click on the Remove From Folder button on the top of the template menu.

3.6 Delete selected document

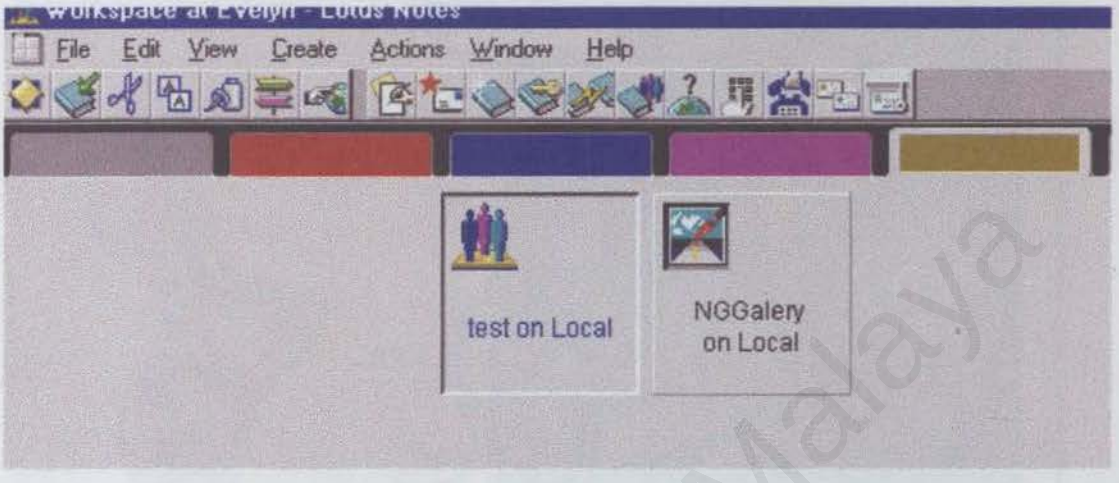
- There is two ways to delete a document. The easiest way to delete the document is to select the document in the template view section, press Delete.
- Another way is open the document and click on the Delete button.
- To confirm the deletion of the document. Press F9 and then click on OK button to confirm. Anyway, confirmation of deletion will pop up before exiting the toolkit.

3.7 Switch to Design view

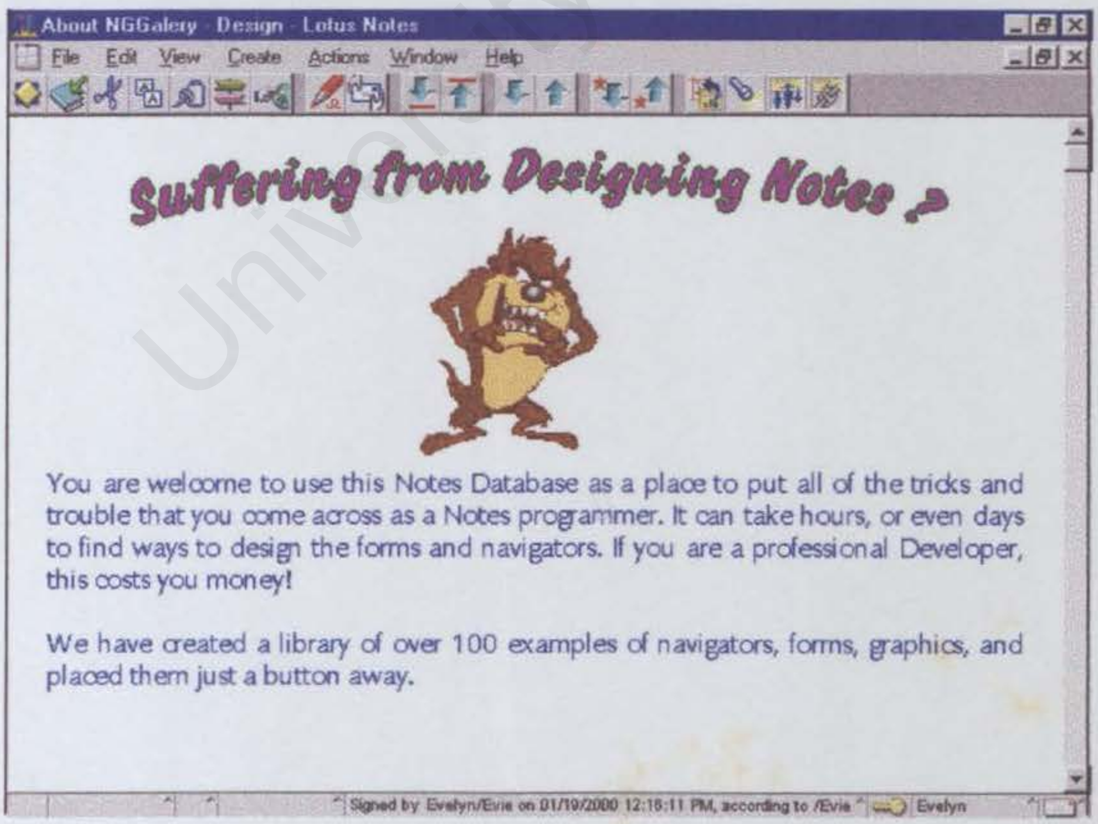
- User may switch to design view by clicking the Switch Design button that appeared in the template menu.
- User may switch back the design view to template menu by clicking the same button.

Appendix E Scenarios

Scenario 1 : Copy graphics from NGG toolkit to a form of another database.


Workspace at Evelyn - Lotus Notes
File Edit View Create Actions Window Help
test on Local
NGGalery on Local

- A database called Test with a blank form as below.
- Get graphics from NGG toolkit. Open NGG toolkit.

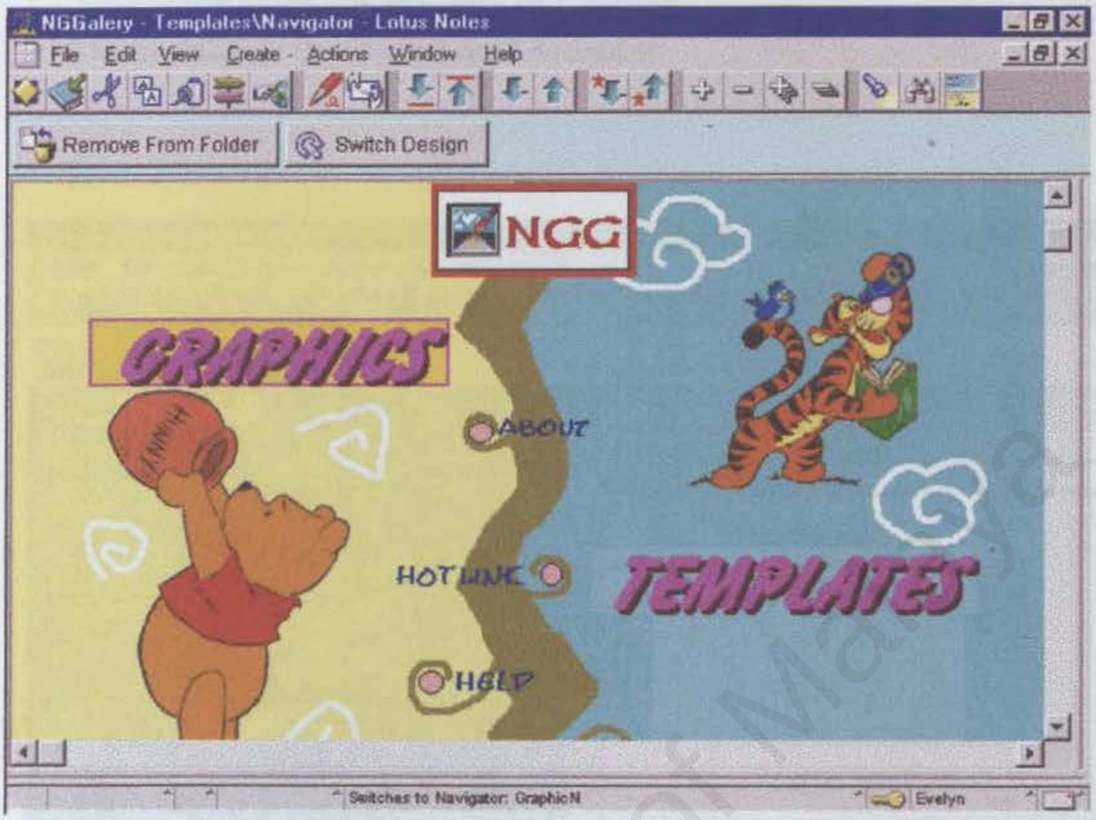

About NGGalery - Design - Lotus Notes
File Edit View Create Actions Window Help
Suffering from Designing Notes ?

You are welcome to use this Notes Database as a place to put all of the tricks and trouble that you come across as a Notes programmer. It can take hours, or even days to find ways to design the forms and navigators. If you are a professional Developer, this costs you money!

We have created a library of over 100 examples of navigators, forms, graphics, and placed them just a button away.

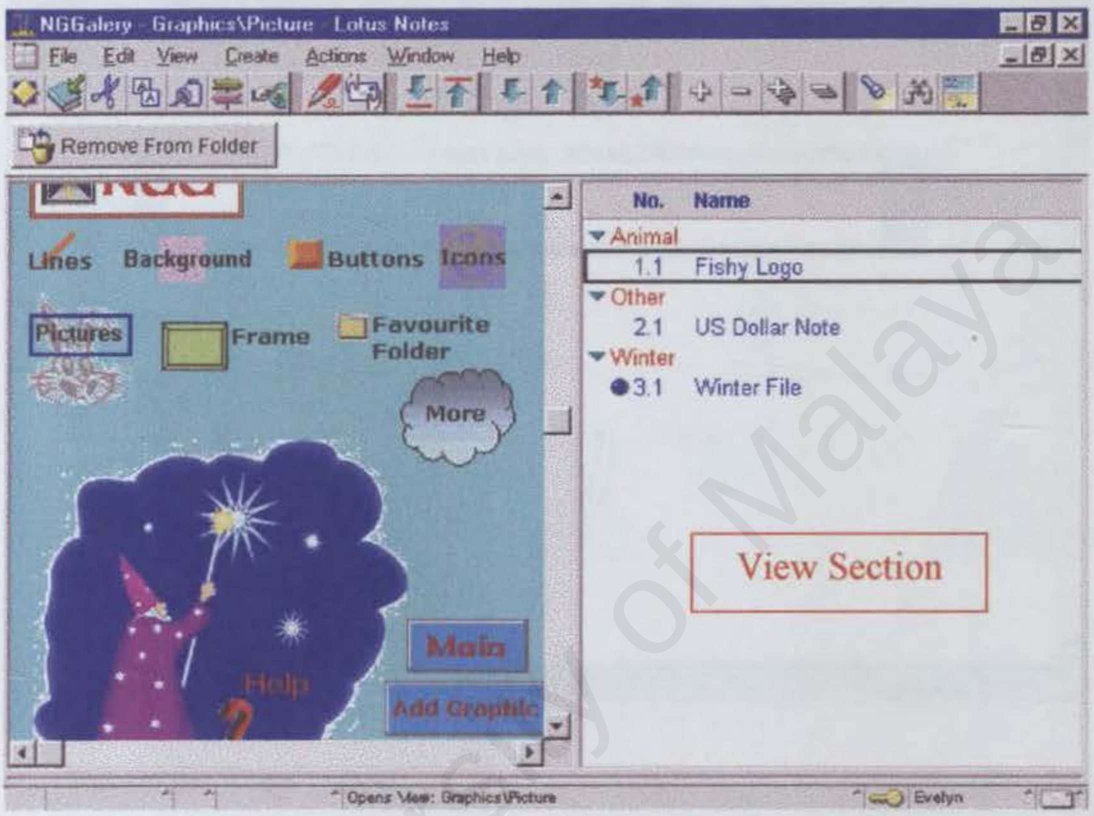
Signed by Evelyn/Evie on 01/19/2000 12:18:11 PM, according to /Evie
Evelyn

- Press Esc to access to Main menu.


NGGalery - Templates\Navigator - Lotus Notes
File Edit View Create Actions Window Help
Remove From Folder
Switch Design
NGG
GRAPHICS
ABOUT
HOTLINK
HELP
TEMPLATES
Switches to Navigator: GraphicN
Evelyn

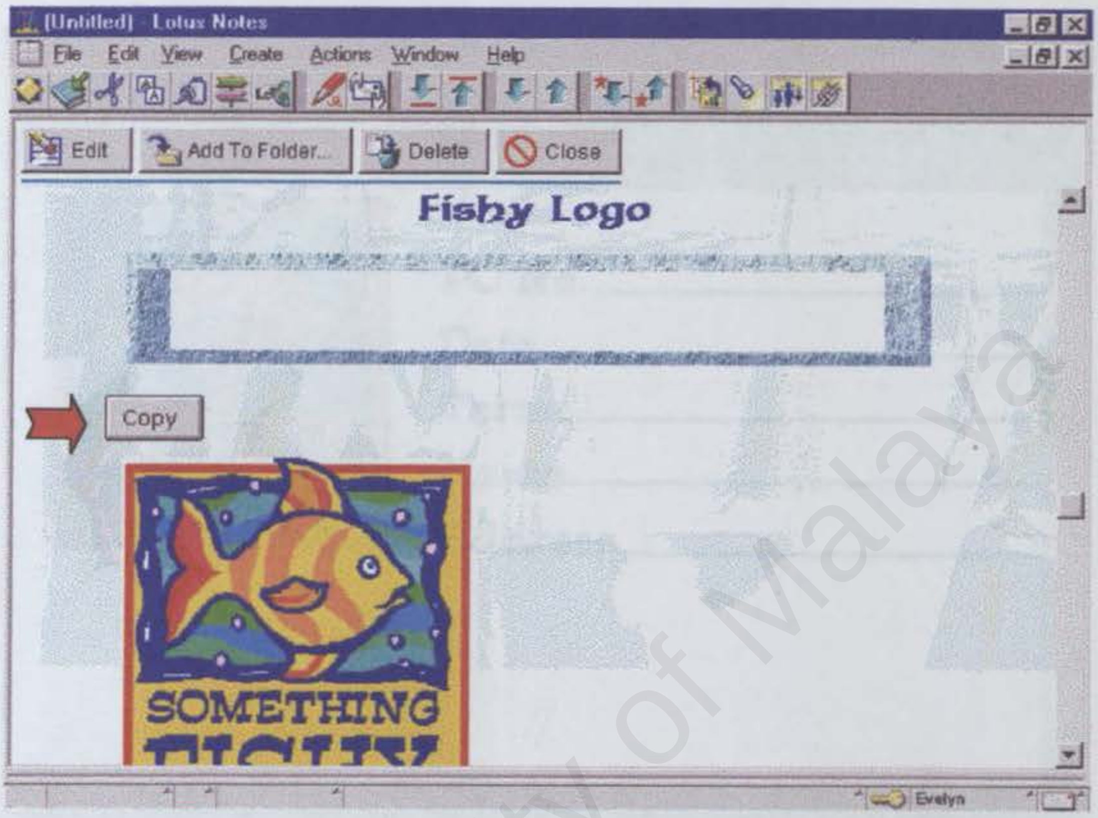
- Click on Graphics hotspot rectangle to access Graphics menu.

- Choose one of the graphics from view section.
- Double-click on the "Fishy Logo" document.

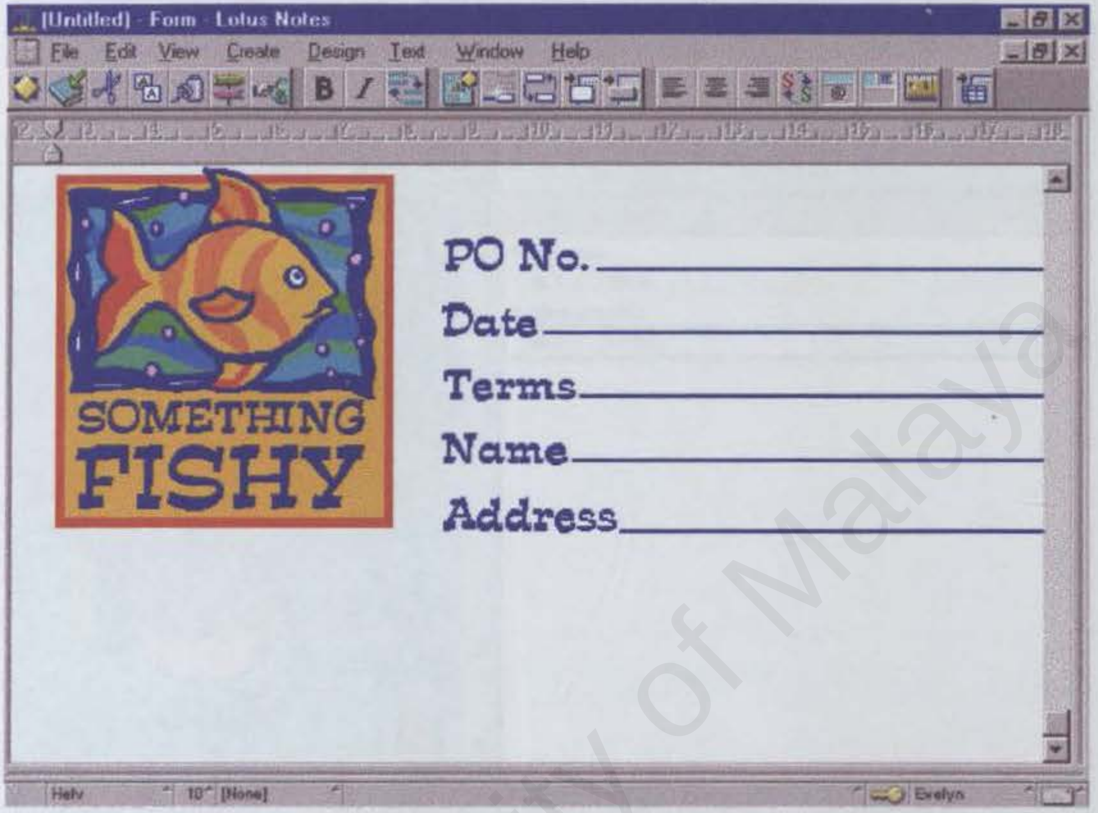

NGGalery - Graphics\Picture - Lotus Notes
File Edit View Create Actions Window Help
Remove From Folder
Lines
Background
Buttons
Icons
Pictures
Frame
Favourite Folder
More
Main
Help
Add Graphic
Opens View: Graphics\Picture
Evelyn

No.	Name
▾ Animal	
1.1	Fishy Logo
▾ Other	
2.1	US Dollar Note
▾ Winter	
●3.1	Winter File

View Section



- Click on the 'Copy' button.
- Go back to 'Test' database and paste on the blank form.
- Steps from the above are repeated to get more graphics.


[Untitled] - Form - Lotus Notes
File Edit View Create Design Text Window Help
SOMETHING FISHY
Helv 10 [None]
Evelyn

PO No.____________________

Date____________________

Terms____________________

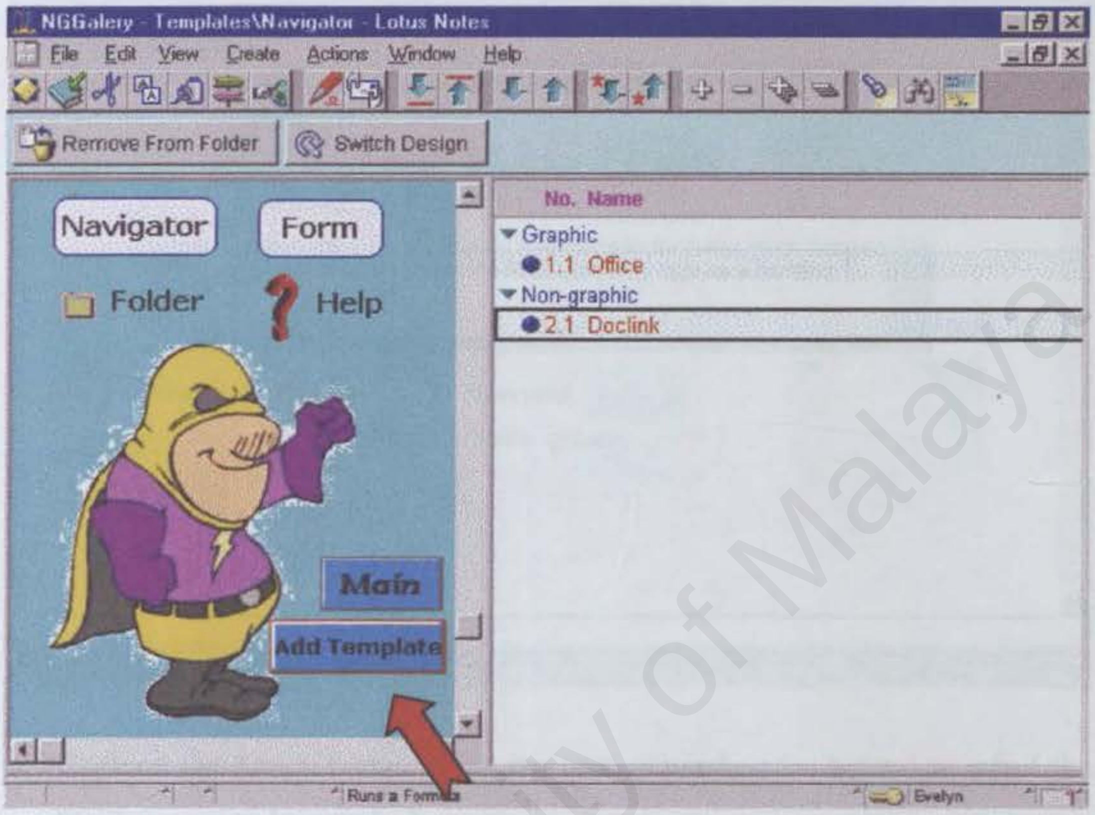
Name____________________

Address____________________

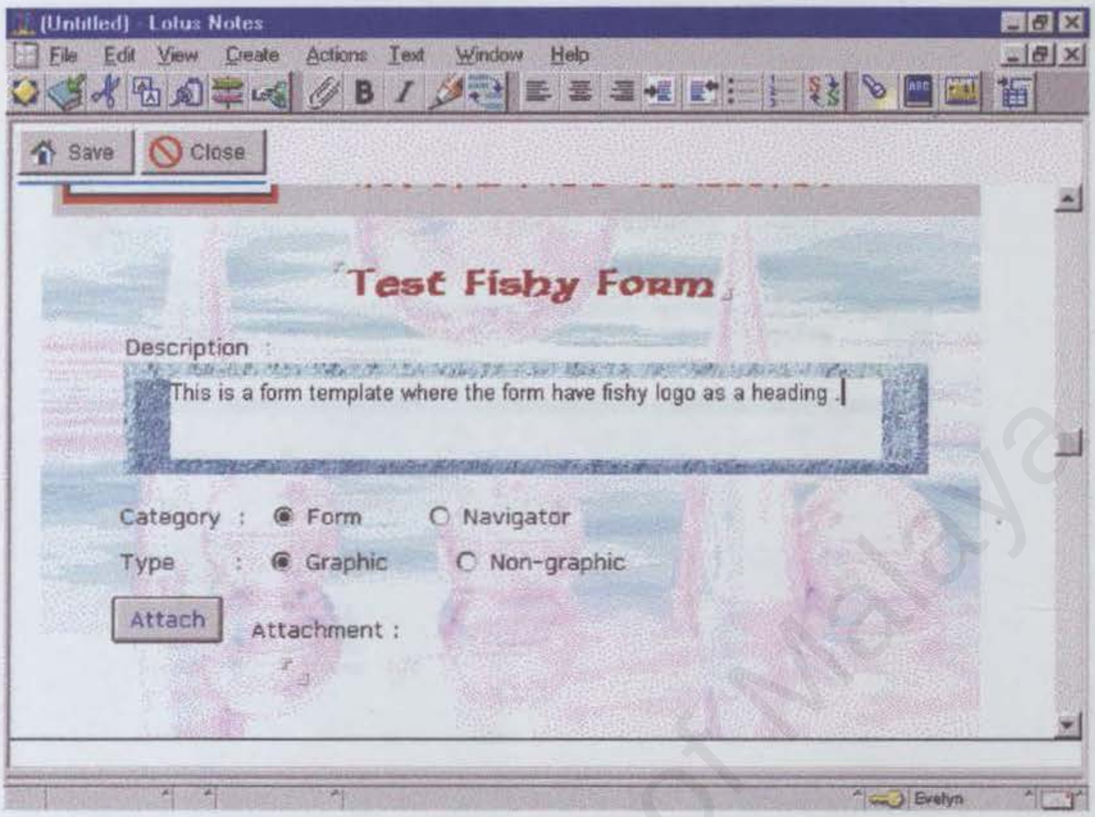
- This is the form with graphics created in the 'Test' database using NGG toolkit.

Scenario 2 : Attach a template database into Template Form.

- Click on the Template hotspot to go to Template menu.



- Fill up the form below.



Test Fishy Form

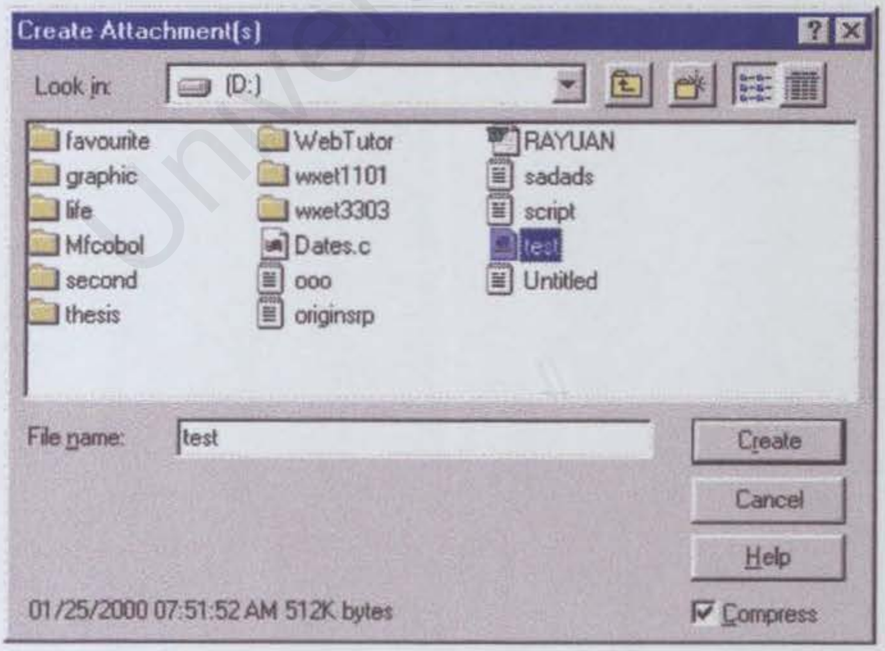
Description : This is a form template where the form have fishy logo as a heading .

Category : ● Form ○ Navigator

Type : ● Graphic ○ Non-graphic

Attach Attachment :

- Click on the Attach button to get the pop up window (as below) to select the file that want to be attached.

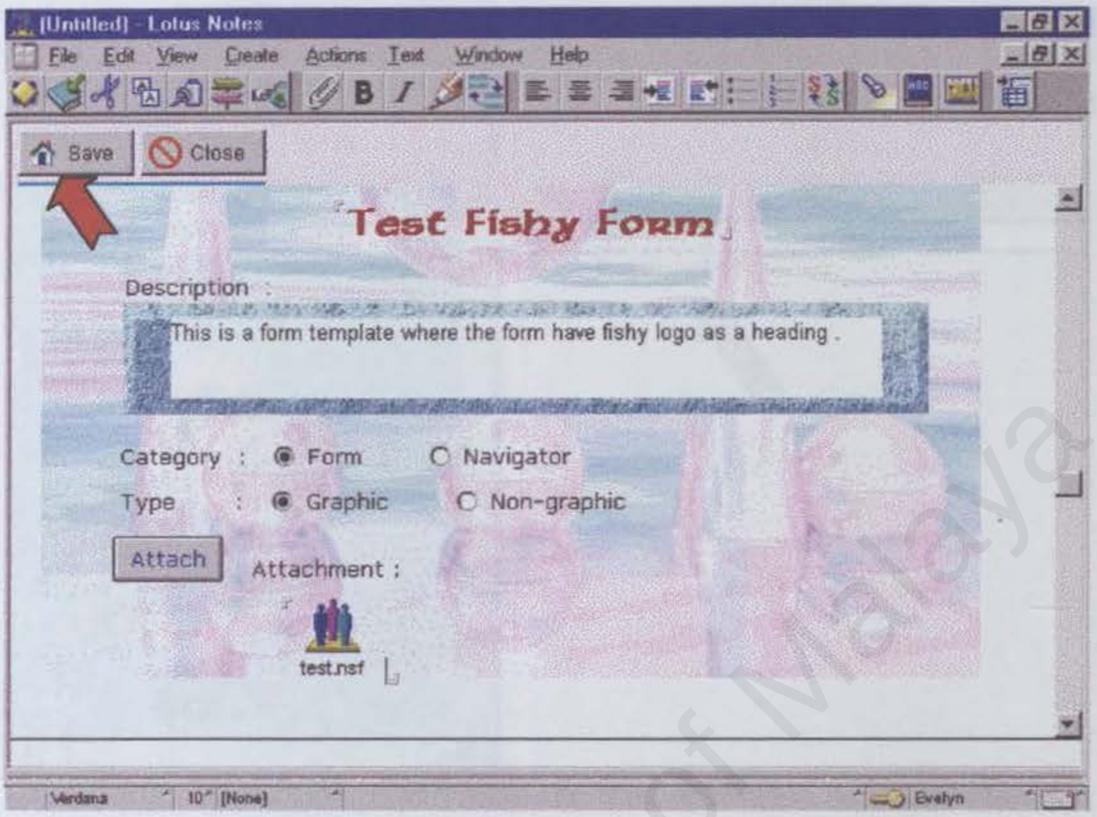


File name: test

01/25/2000 07:51:52 AM 512K bytes

☑ Compress

- Click Save button on the top of the form to save the document.



Test Fishy Form

Description :

This is a form template where the form have fishy logo as a heading .

Category : ● Form ○ Navigator

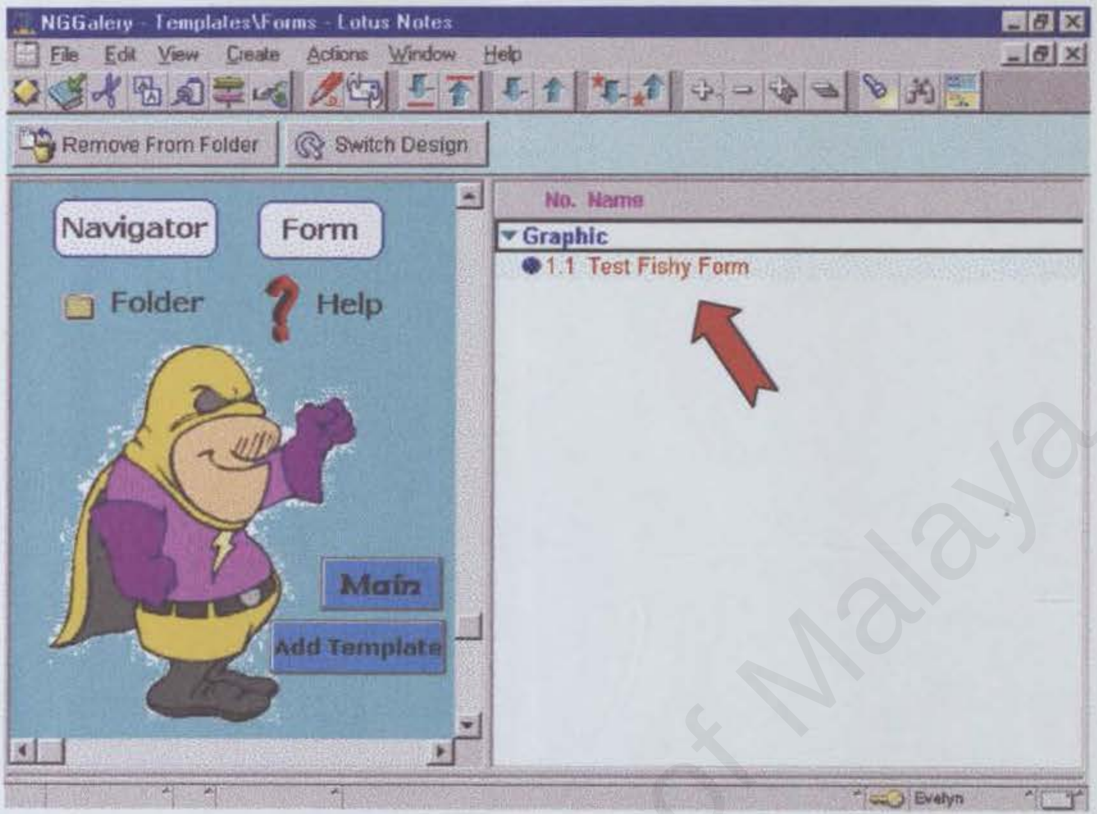
Type : ● Graphic ○ Non-graphic

Attach

Attachment :

test.nsf

- Click on Close button to get back to the Template menu.



- Now the attachment has been save into a document named 'Test Fishy Logo' and it can be seen in the view section.