Bilkent University Computer Science Department

Object Oriented Software Engineering

CS-319

Summer Term Project Design Report

Dear Diary

- Abdurrezak Efe 21301883
- Ayşegül Sümeyye Kütük 20900538
- Enes Kavak 21302618

1. Introduction

1.1 Purpose of the system

Dear Diary is a neat desktop application that offers its users a diary experience via an elegant interface allowing to keep the archive of writings, albums, special days/dates through the different functions of the application software structure. Comparing to the other diary applications in the market, Dear Diary looks a lot fancier, works with a more effective environment and offers a more practical usage to the candidate diary writers. Within the project improvement, we aim to encourage people to share their experiences in a reliable environment and to make their own personal journal of daily events, appointments, secrets and feelings. Through the practical usage of the application system, Dear Diary serves as a tool for its users to acquire the habit of writing daily diary.

1.2 Design Goals

To determine the design goals is a must in terms of building the details of the system in order to declare the application qualities. Hence, we made quite an effort to build our design goals on the functional and non-functional requirements of the system that we summed elaborately in the analysis report. Our specific design goals are presented below.

End User Criteria:

Ease of Use: Since it is one of our goals to offer a practical usage and encourage people to write more, we tried to build an application software with a neat

and simple design. In terms of usage, our application interface provide its users a distinct menu by which the users will easily navigate themselves through the desired operations they wish to fulfill. Since all the operation transitions are depending on the user mouse input, the usage of the system is quite explicit.

Ease of Learning: Since we present a brand new application, the operation flow of the system will be unknown for the user point of view. The buttons and the menu details are assigned with reasonable tags. With that, we try to help the users to understand which button will generate which operation. Since the user interface have become quite clear to follow by this way, we thought it would be trivial to add an instructive user guide.

Maintenance Criteria:

2. Software Architecture

2.1. Overview

In this part, we will try to explain how we decomposed the software we develop into subsystems in order to make it robust and efficient. As there is no multilayer structure in our project that needs a special kind of design pattern, our design is neat and straightforward.

2.2. Architectural Styles

2.2.1 Layers

The architectural structure of Dear Diary possesses three layers. Namely:

- User Interface
- Diary Manager
- Diary Entities

Our system is user dependent which implies that the first layer that interacts with lower layers of the program is the user interface. We intended to provide our users with a fancy and easy-to-use interface.

The second layer is Diary Manager which controls almost everything. Although is not coded in a new class, Diary Manager controls every delete, save, update operation there is. However, it needs another sublayer which is Diary Entities layer.

Diary entities layer is the lowest level of abstraction in our program and makes every operation possible by providing corresponding entities and objects.

2.2.2 Model View Controller

By using Model View Controller model, we made it easier for the program to perform its tasks. As MVC proposes a system that has three basic subsystems responsible for the data, its processing and its displayment in a desired way.

In our system, the model has the raw data of users, days, entries, albums and everyting Dear Diary stores and processes. Meanwhile, view displays the data that is being modified or updated in runtime. Finally, controller manipulates the model data to update the system everytime the user interacts with the system.

2.3. Hardware / Software Mapping

Our program has been being implemented in Java programming languase so it uses the latest JDK. Meanwhile, Dear Diary interacts with users by a keyboard to write entries, notes, album descriptions etc. and mouse to click on buttons.

The system's requirements are minimal as it is a small sized desktop application with data stored in .txt and .ser files.

As Java provides its developers with Serializable interface for objects to be stored in .ser files, we had no hardship in storing objects in .ser files with security.

Our system can easily be launched in any operating system as long as the OS has an up-to-date Java compiler. Finally, our program does not need any internet connection to work properly.

2.4. Persistent Data Management

As Dear Diary is not a very advanced based complex project, we needed no well designed databases or any advanced data structure to store the data of the users and its sub files. Thus, Dear Diary stores all the data it has in .txt files which keep track of logins, signups, dates of signups and logins; meanwhile, .ser files store the updated data of any user and its components.

When Dear Diary is launched, no file is on use as long as no sign up or login operation is performed. When a login/signup is done, .txt files make sure that the new user has a valid username with a sufficient password or if the user entered his password correctly to enter the system.

After a successfull login operation, a particular .ser file is in use. However, as there is no way for Java forms to get information from another, the correspondng .ser file is processed in every Java form once they are opened.

Whenever an addition, save, deletion or update operation is performed; the corresponding .ser file is updated accordingly in runtime.

2.5. Access Control and Security

Dear Diary application, as the name proposes, needs a strong password protection and data privacy. Therefore, the users will not be able to see each others' entries, notes, talks, albums and not even days. To provide users with privacy, our system dictates that the user has a minimum seven characters long password with a unique username. Actually, the mentioned signup, login structure was the hardest part to implement and provide the users with. Nevertheless, a diary with no protection of the data is as useles as a bank with no gate.

Additionally, as our system does not have any kind of third party connection such as internet, bluetooth etc. it is not vulnerable to any attack.

2.6. Boundary Conditions

Initialization

The program needs no installation, it can easily be launched from .java file using a Java compiler. When launched, the program will handle the rest by itself.

Termination

Dear Diary can be terminated by x icon on top right/left corner(depending on the OS it is run on). Since the data is updated in runtime, no data loss is of concern. However, as mentioned before the data that was not saved will not be of use in future if the form is disposed by the user.

Additionally, there is also a logout button in every frame of Dear Diary which simply return to the login page.

Error

As the program is coded with Java, it handles exceptions very well. Whenever, a file is corrupted the system simply do not load the data stored in that file. On the other hand, although the program is robust enough to be run many times consecutively, cannot save data in some situations like non-responding frames due to RAM's situation.