Photonest

Object Design

1.0

<Date>

<Feyzullah Berkay Danış

Özay Ezerceli

Nursena Karakulah

Dilara Ünbay>

Prepared for

SE302 Software Engineering



**OBJECT DESIGN DOCUMENT**

Object Design Document (ODD) describes object design trade-offs made by developers, guidelines they followed for subsystem interfaces, the decomposition of subsystems into packages and classes, and the class interfaces. The ODD is **used** to exchange interface information among teams and **as a reference during testing**. The audience for the ODD includes system architects (i.e., the developers who participate in the system design), developers who implement each subsystem, and testers.

Among three approaches to generate ODD, we follow “**ODD embedded into source code**” approach in SE301, since the other methods create many redundancies, inconsistencies.

The initial version of the ODD can be written soon after the subsystem decomposition is stable. Both packages and class interfaces can be generated from source code (comments!) by using a tool, which is named Javadoc. Keeping material for the ODD with the source code enables the developers to maintain consistency more easily and rapidly.

1. **Introduction**

Describes the general trade-offs made by developers (e.g., buy vs. build, memory space vs. response time), guidelines and conventions (e.g., naming conventions, boundary cases, exception handling mechanisms), and an overview of the document. Interface documentation guidelines and coding conventions are the single most important factor that can improve communication between developers during object design. These include a list of rules that developers should use when designing and naming interfaces.

**1.1.Object Design Trade-offs**

During the object design of our project we had make several design trade-offs. One of them was deciding between writing the code ourselves or outsource it. We have written almost all the code by ourselves, we used firebase methods during authentication, change password and forgot password functions.

Another trade-off decision we made was between memory space and response time of the application, since we wanted to make this application enjoyable by users we decided to choose response time over memory space because in a mobile application like Photonest a slow response time could make user dislike the app.

We had to decide between delivery time and functionality and we decided to give delivery time more priority. Since we had a certain delivery date, we decided to add as much functionality to the system as we can and deliver it on time.

**1.2.Interface Documentation Guidelines**

In order to create better communication between developers during object design we developed some guidelines for interface documentation. We used singular nouns or phrases for class names that included activity or fragment at the end of the name to indicate weather the class is an activity or fragment and every word in the name started with a capital letter. All the objects created were derived from the class diagram and use cases in the RAD. When an exception is caught, the error is displayed to the user.

**1.3.Definitions, Acronyms, and Abbreviations**

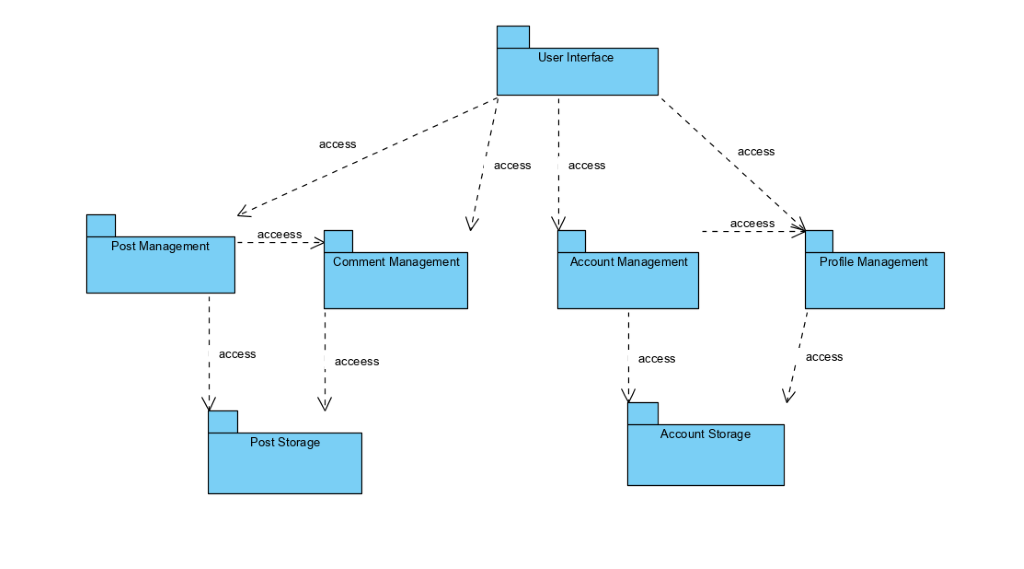
RAD : Requirement Analysis Document

**1.4.References**

* Instagram

1. **Packages**

Describes the decomposition of subsystems into packages and the **file organization of the code.** This includes an overview of each package, its dependencies with other packages, and its expected usage.



* 1. **User Interface Package Definition**

**2.1.1 File organization of code for User Interface Package**

This package contains the following files,

res/

,which this file includes the files,

* + anim/ : This file contains the xml files for animations needed in package.
  + drawable/ : This file contains the xml/png/jpeg files for icons and pictures needed in the package.
  + layout/ : This file contains xml files for templates of pages needed in the package.
  + menu/ : This file contains xml files of menu designs’ needed in the package.
  + mipmap/ : This file contains design files of launcher icon.
  + values/ : This file contains references to reach values of multiple used string, color, styles.

**2.1.2 Overview of User Interface package**

This package is responsible of showing reactions received from controller subsystem after the user performed a function. Any data in the model subsystem will be displayed to the user through this package.

**2.1.3 Dependencies with other packages**

User interface package contains layout files that are independent from other packages, this package will display reactions received from other packages.

**2.1.4 Usage of User Interface package**

User interface package will contain the entire GUI and it will provide design layout of every page, activity and fragment of the application that user interacts with. Also it will provide images and icons that were used in application which makes the application visually interesting, menus that users use to navigate through the activities and packages.

* 1. **Profile Management Package**

**2.2.1 File Organization of code for Profile Management Package**

This package will include the following files:

* + com.302.photonest/

This file contains activity and fragment classes that has the functions provided by the Profile management package in them.

**2.2.2 Overview of Profile Management Package**

The Profile Management Package is responsible with handling functions that are related with profile information. It retrieves all user information from firebase database.

**2.2.3 Dependencies with other Packages**

The Profile Management Package is related with Account Management and User Interface. When a user needs to be reached from firebase database user information is first received from Account Management. When a user wants to perform a function provided by this package user interacts with the function through the User Interface Package.

**2.2.4 Usage of the Package**

Profile Management Package is used for user profile functions such as edit profile, change profile photo, delete profile photo. This package is also used to retrieve user data from fireabase and display it to user.

* 1. **Account Management Package**

**2.3.1 File Organization of code for Account Management Package**

This package will contain the following files:

* + com.302.photonest/

This file includes activity and fragment classes that contain code for functions provided by this package.

**2.3.2 Overview of Account Management Package**

Account management handles authentication and other account related functions such as register, login, forgot password, change password logout. It retrieves user’s account information from Account Storage Package.

**2.3.3 Dependencies with other Packages**

Account Management package is dependent on the User Interface Package and Account Storage Package. Users interact with the functions through User Interface Package and Account Management Package retrieves user account information from Account Storage.

**2.3.4 Usage of the Package**

Account Management Package handles account related functions, it retrieves user input from User Interface package and updates data of Account Storage Package and perform appropriate action according to user input.

* 1. **Post Management Package**

**2.4.1 File Organization of code for Post Management Package**

Post Management Package contains the following files,

* + com.302.photonest/

This file contains activity and fragment classes that contains functions provided by this package.

**2.4.2 Overview of Post Management Package**

Post Management Package handles functions related to posts and applies changes to post data on firebase realtime database.

**2.4.3 Dependencies with other packages**

Post Management package is dependent on User Interface Package and Post Storage Package. It retrieves user input from User Interface to perform appropriate function and accesses post data from Post Storage Package.

**2.4.4 Usage of the Package**

Post Management Package performs functions of creating, deleting and editting posts. It receives the required post data coming from user through the User Interface package and updates the post data on Post Storage Package.

* 1. **Comment Management Package**

**2.5.1 File Organization of code for Comment Management Package**

Comment Management Package contains the following files,

* + com.302.photonest/

This file contains activity and fragment classes that contains functions provided by this package.

**2.5.2 Overview of Comment Management Package**

Comment Management Package handles comment related functions and accesses to comment data on Post Storage Package.

**2.5.3 Dependencies with other Packages**

Comment Management Package dependent on Post Storage, User Interface and Post Management. Comment Management Package receives comment data from Post Storage Package, it receives user input from User Interface Package before applying appropriate function, while using functions provided by the package comments reaches to post data through Post Management Package.

**2.5.4 Usage of the Package**

Comment Management Package handles comment related functions such as adding or deleting comment.

* 1. **Post Storage Package**

**2.6.1 File Organization of code for Post Storage Package**

Files related to this package does not exist in application they are held on firebase realtime database.

**2.6.2 Overview of Post Storage Package**

Post Storage Package holds data about posts and comments that are added by users. Application acceses this package for post and data.

**2.6.3 Dependencies with other packages**

Post Storage Package is dependent on Comment Management Package and Post Management Package. Post Storage Package receives input from Comment Management before updating comment data and receives input from Post Management before updating post data.

**2.6.4 Usage of the Package**

This package holds all the information about posts and comments, other packages reaches to Post Storage Package for post and comment data.

* 1. **Account Storage Package**

**2.7.1 File Organization of code for Account Storage Package**

Files related to this package does not exist in application they are held on firebase realtime database.

**2.7.2 Overview of Account Storage Package**

Account Storage Package holds data about user accounts and profiles that are created by users such as username, email, password-are kept as encrypted-, full name.

**2.7.3 Dependencies with other Packages**

Account Storage package is dependent on Profile Management and Account Management packages. This package receives input from Profile Management Package before updating profile data of user, it receives input from Account Management Package before updating account data.

**2.7.4 Usage of the Package**

Account Storage is used for holding profile and account related data. When a data about profile or account needs to be reached other packages access this Package.

1. **Class Interfaces**

Describes the classes and their public interfaces. This includes an overview of each class, its dependencies with other classes and packages, its public attributes, operations, and the exceptions they can raise.