Deliverable #2 Template

SE 3A04: Software Design II – Large System Design

1 Introduction

1.1 Purpose

The following document will outline the user interactions, architecture, and classes of the application, Nature Optix. The intended audience for this document will be the teaching assistants as well as Dr. Ridha Kedhri. It is the goal of this document to provide insight to the intended audience into the internal mechanisms of Nature Optix. After reading this document, the audience should have a good understanding of the way in which a user can interact with the system, the architecture that the system uses, and the classes that comprise the system.

1.2 System Description

The Nature Optix application will allow the user to identify a natural phenomena. The application will ask the user a series of questions and the user will be able to specify an answer for each question from a drop down menu. The application will use a combination of repository and blackboard architecture. It will be comprised of 6 entities; 3 experts (a physicist, a field expert, and a meteorologist), a location expert, a database of natural phenomena, and a controller. The database will contain all phenomena and information about the characteristics of each phenomena. Each expert will act as an algorithm that compares a characteristics of phenomena in the database to the user input. The physicist will compare the visual attributes, the field expert will compare the physical attributes, and the meteorologist will compare the weather condition attributes required for each phenomena to occur. Each expert will create a list of phenomena that has characteristics that match the user input. The three lists will then be sent to the controller. The controller will determine what phenomenon is common to all three lists and will present this phenomenon to the user. The experts will not be able to communicate to each other and will only have access to the database and controller.

1.3 Overview

The user interactions will be shown using a use case diagram, along with descriptions of each interaction. The architecture for the system will be presented using descriptions along with justifications for decisions. Furthermore, the classes of the application will be outlined using an analysis class diagram as well as a class responsibility collaboration.

2 Use Case Diagram

1. User wishes to access the app. The user shall be able to download the app onto their smart phone in order to access it. This downloading action will be handled by the operating system. User shall be able to open the app. The system developer needs to ensure that application shall be able to handle user input, so that the user can access and interact with the app.

2. User wishes to take a picture and post it to social media. User can then access the built-in camera on phone and it's functionality from the app. User can also save the picture to the app and the phone if they so desire. User shall be able to post to social media directly from the app. The system developer shall ensure that application can access the internet via wireless connection from smart phone, the

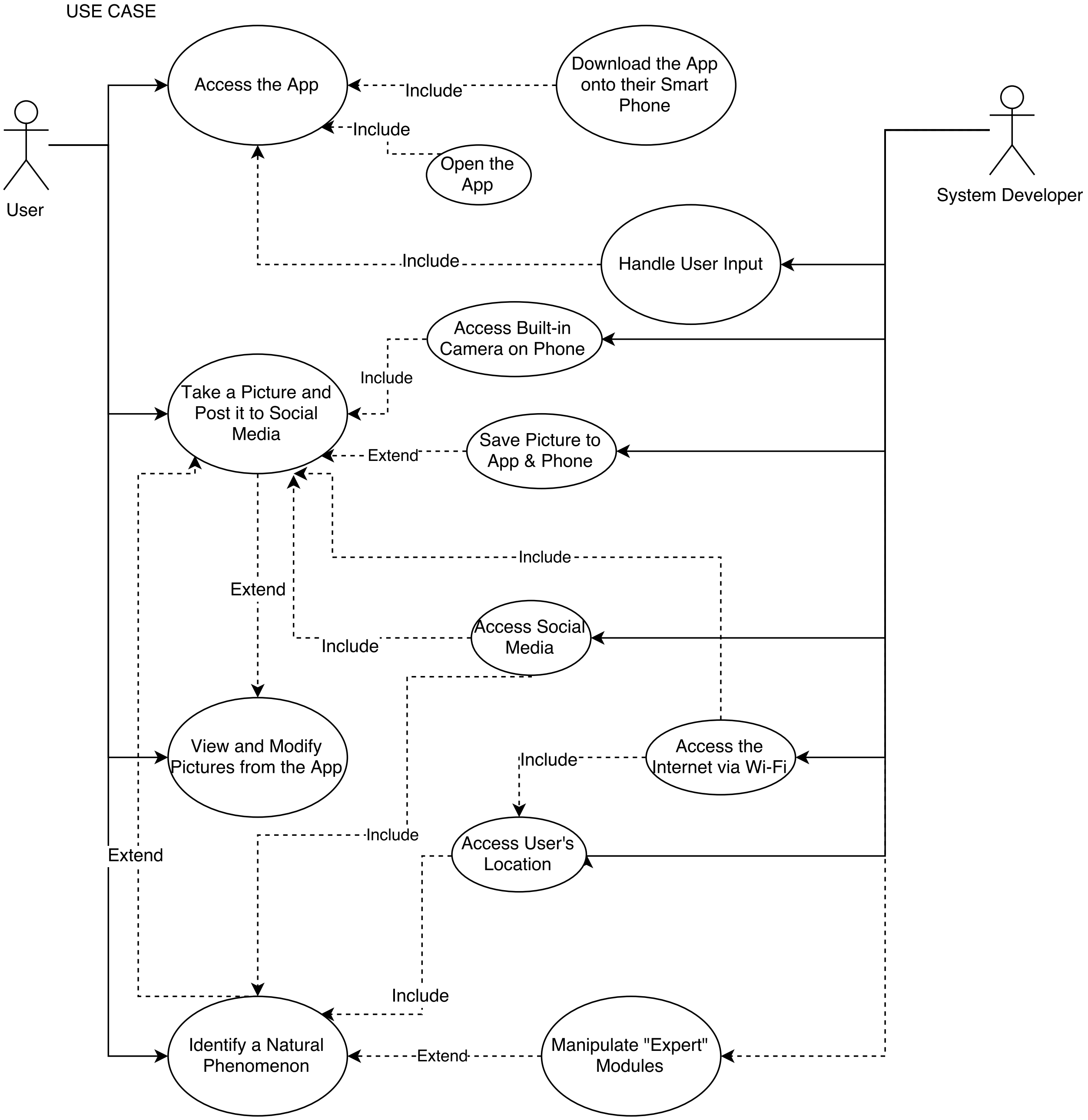
USE CASE
User
System Developer
Access the App
Download the App onto their Smart Phone
Include
Open the App
Include
Handle User Input
Include
Access Built-in Camera on Phone
Include
Take a Picture and Post it to Social Media
Save Picture to App & Phone
Extend
Include
Extend
Access Social Media
Include
View and Modify Pictures from the App
Access the Internet via Wi-Fi
Include
Access User's Location
Include
Extend
Include
Identify a Natural Phenomenon
Manipulate "Expert" Modules
Extend

Figure 1: Use Case Diagram

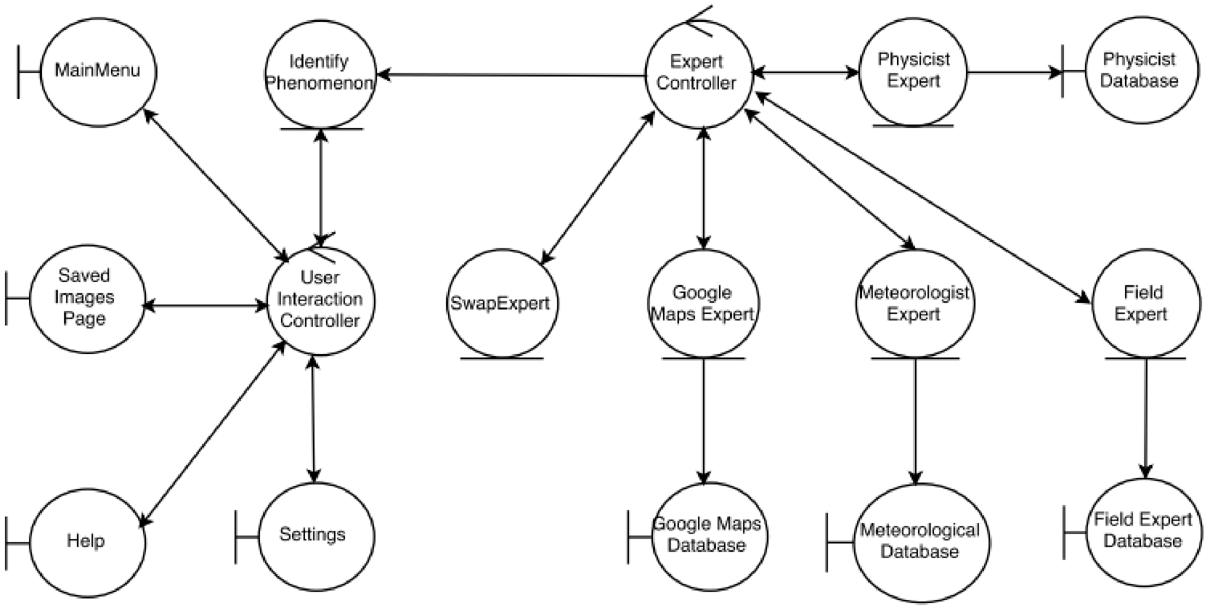

Figure 2: Analysis Class Diagram

built-in camera on the smart phone, and social media (Instagram). Application shall be able to save pictures directly to the phone.

3. User wishes to view and modify pictures from the app. User shall be able to view saved pictures through the app and the phone. User shall be able delete pictures from the app and the phone. System Developer shall ensure that application can display requested pictures to the user. Application shall be able to delete pictures directly on the phone.

4. User wishes to identify a natural phenomena. User shall be able to post a picture on social media through the app. The system developer shall ensure that the app can access the user's location using google maps services. App shall have access to the internet via wireless connection from smart phone. App shall be able to switch ”expert” modules in order to identify the natural phenomenon.

3 Analysis Class Diagram

4 Architectural Design

This section should provide an overview of the overall architectural design of your application. You overall architecture should show the division of the system into subsystems with high cohesion and low coupling.

4.1 System Architecture

a) Identify and explain the overall architecture of your system

b) We are using the Model-View-Controller architecture for our system.

c) The Model View Controller hierarchy helps distinguish the business logic from the controller logic and is useful in creating modularity within the program. We are looking at this from 2 viewpoints:

d) Our application is centered around displaying information to the user after they provide us with an initial input; the client will be focused on the view after the initial query which is handled by the model and finally connected by the controller to the view. Hence, an MVC architecture seems useful.

e) MVC sets up our application to be enhanced further upon as it facilitates modular, high cohesion, low coupling design. We can add new modules without disrupting the existing codebase.

f) Provide a structural architecture diagram showing the relationship among the subsystems (if appropriate)

4.2 Subsystems

NatureOptix will be broken down into six subsystems. These subsystems are

1. Controller
2. Experts x4,
3. Database.

The first Controller is the subsystem to controlata between the user and the experts where the data goes. The controller will ask the user a series of questions. It will then take this information and pass it to the four experts. After the experts have returned with the potential answers to determine what phenomenon the user might be seeing, the controller will then present the options to the user.

Three of the four Experts will be implemented as algorithms. These three experts are the Physicist, the FieldExpert, and the Meterologist. The experts will answer questions that pretain to their field. The Physicist will handle all physical aspects of natural phenomeon. This includes colour, shape, and size. The FieldExpert will be the expert responisble for identification based on attributes such as temperature and humidity. Finally the Meterologist will attempt to identify phenomeon based on the current weather at the location. The location expert will be the Google Maps API.

The final subsystem is the Database. Each expert will be assigned a Database. This ensures that the experts are easily swappable. The Database will store all the researched data about the natural phenomenon. Each expert will access the assigned database when needed. The second Controller will

5 Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

Class Name: Database	
Responsibility:	**Collaborators:**
Has information on all the phenomenons and each of their attributes.	

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

Class Name: Expert1:Physicist	
Responsibility:	**Collaborators:**
For attributes shape, colour, opacity, size, angle, elevation and brightness it will compare the attribute specifications given by the user to the phenomenons in the database and create a list of phenomenons that match the user input.	Database, Controller

Class Name: Expert2:FieldExpert	
Responsibility:	**Collaborators:**
For attributes temperature, density and moistness it will compare the attribute specifications given by the user to the phenomenons in the database and create a list of phenomenons that match the user input.	

IMPORTANT NOTES

- Please document any non-standard notations that you may have used
  - *Rule of Thumb*: if you feel there is any doubt surrounding the meaning of your notations, document them
- Some diagrams may be difficult to fit into one page
  - It is OK if the text is small but please ensure that it is readable when printed
  - If you need to break a diagram onto multiple pages, please adopt a system of doing so and thoroughly explain how it can be reconnected from one page to the next; if you are unsure about this, please ask about it
- Please submit the latest version of Deliverable 1 with Deliverable 2
  - It does not have to be a freshly printed version; the latest marked version is OK
- If you do NOT have a Division of Labour sheet, your deliverable will NOT be marked

Class Name: Expert3:Meterologist	
Responsibility:	**Collaborators:**
For attributes sun, temperature and precipitation it will compare the attribute specifications given by the user to the phenomenons in the database and create a list of phenomenons that match the user input.	

Class Name: Controller	
Responsibility:	**Collaborators:**
Gives experts access to the database, and determines common phenomenons amongst the lists of the three experts and returns phenomenons to the user.	Expert1, Expert2, Expert3, GoogleAPI, Database

Class Name: GoogleAPI	
Responsibility:	**Collaborators:**
Determines users location to create list of phenomenons that could occur in the area that the user is in by comparing the location information to the location attribute in the database of phenomenons.	Database, Controller