

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, NatureOptix. 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 NatureOptix. 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 NatureOptix 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 9 entities; 3 experts (a physicist, a field expert, and a meteorologist), a location expert, a database corresponding to attributes identified by each one of the experts for every optical natural phenomena, and a controller. Each database will contain information about each phenomena specific to what it's expert knows. Each expert will act as an algorithm that compares characteristics of a 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. In some boundary cases, the applications shall provide the users with an aggregated list of closest match results to their query. The experts will not be able to communicate to each other and will only be communicating with 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

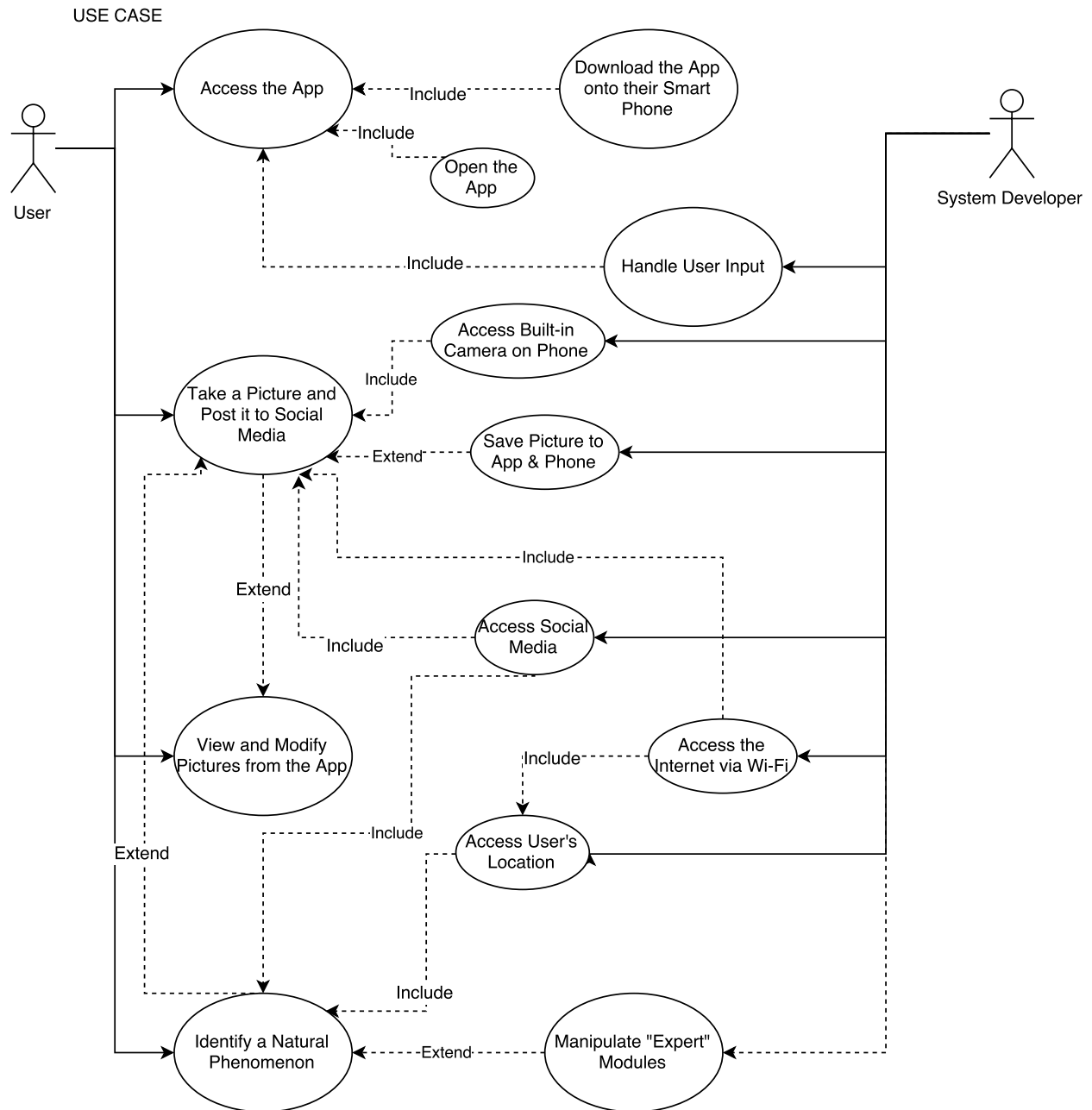


Figure 1: Use Case Diagram

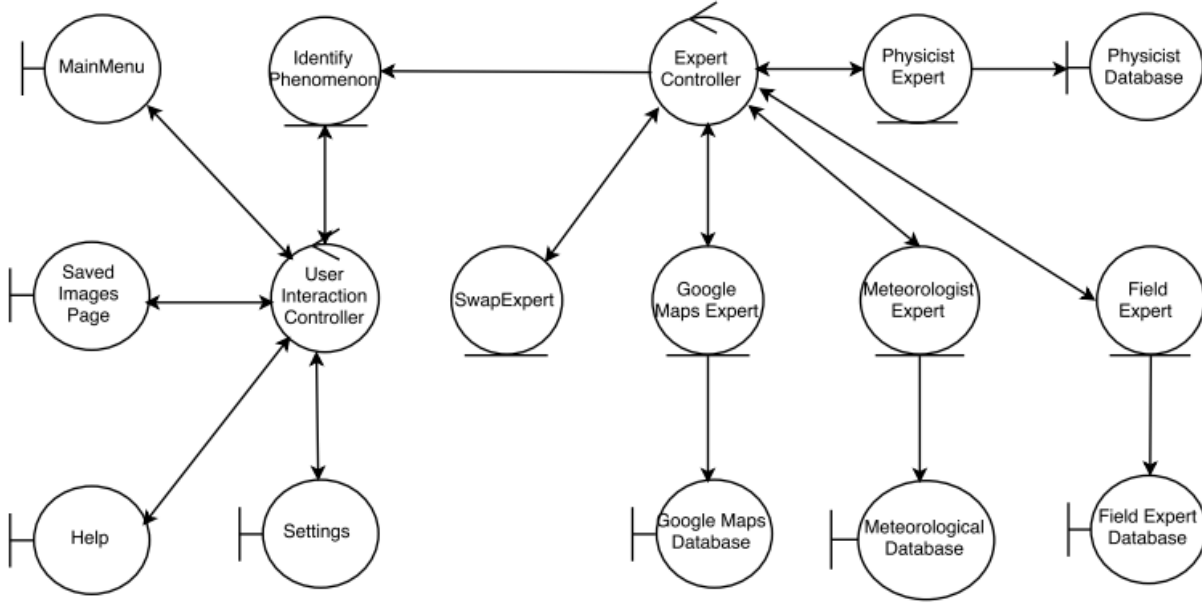


Figure 2: Analysis Class Diagram

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 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. Your overall architecture should show the division of the system into subsystems with high cohesion and low coupling.

4.1 System Architecture

- 1) We are using the Model-View-Controller architecture for our system.
- 2) 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:
 - (a) 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.

- (b) 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.

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

4.2 Subsystems

NatureOptix will be broken down into ten subsystems. These subsystems are:

1. UserInteractionController
2. ExpertController
3. Experts (4),
4. Database (4).

The UserInteractionController will ensure that the proper actions occur after the user has selected them. This means that this class will facilitate user experience. The UserInterfaceController will initiate the process of identifying the phenomenon. Once the user has selected that they wish to identify a phenomenon, this controller will send information to the IdentifyPhenomenon module. It will also receive the relevant information from the IdentifyPhenomenon class and pass it along to the user.

The ExpertController will control the experts. It is responsible for passing data to the experts and to pass information to the IdentifyPhenomenon module. The ExpertController is also responsible for facilitating the process of swapping experts. This is done with a back and forth with the SwapExpertClass

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

The final subsystems are the Databases. 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. Each database will have its own schema. This makes having multiple databases necessary.

5 Class Responsibility Collaboration (CRC) Cards

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: Physicist Database | |
|--|------------------|
| Responsibility: | Collaborators: |
| Has information on all the attributes of each phenomenon that the physicist would be looking for. This includes shape, colour, opacity, size, angle, elevation and brightness. | Physicist Expert |

| Class Name: Field Expert Database | |
|--|----------------|
| Responsibility: | Collaborators: |
| Has information on all the attributes of each phenomenon that the field expert would be looking for. This includes temperature, density and moistness. | Field Expert |

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: Meteorologist Database | |
|--|-----------------------|
| Responsibility: | Collaborators: |
| Has information on all the attributes of each phenomenon that the meteorologist would be looking for. This includes sun, temperature and precipitation | Meteorologist Expert |

| Class Name: Google Maps Database | |
|--|-----------------------|
| Responsibility: | Collaborators: |
| Has information on the locations of each phenomenon in order for the Google expert to use. | Google Maps Expert |

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

| Class Name: Field Expert | |
|---|--|
| 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. | Field Expert Database, Expert Controller |

| Class Name: Meterologist Expert | |
|---|--|
| 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. | Meterologist Database, Expert Controller |

| Class Name: Google Maps Expert | |
|--|---|
| Responsibility: | Collaborators: |
| For location attribute it will compare the user's location to the phenomenons in the google maps database and create a list of phenomenons that could be in the user's area. | Google Maps Database, Expert Controller |

| Class Name: Swap Expert | |
|---|---|
| Responsibility: | Collaborators: |
| Switches experts for expert controller to analyse each experts results. | Google Maps Database, Expert Controller |

| Class Name: Expert Controller | |
|--|---|
| Responsibility: | Collaborators: |
| Gives experts information from user and gives the user information complied by experts to identify phenomenon. Also, chooses phenomena that appear in all lists given by experts and sends to user as identified phenomenon. | Identify Phenomenon, SwapExpert, Google Maps Expert, Meteorologist Expert, Field Expert, Physicist Expert |

| Class Name: Identify Phenomenon | |
|--|--|
| Responsibility: | Collaborators: |
| Holds all attributes specified by the user for the Expert Controller. Expert Controller then returns phenomenon identified by experts to this class to be shown to user. | User Interaction Controller, Expert Controller |

| Class Name: User Interaction Controller | |
|--|--|
| Responsibility: | Collaborators: |
| Coordinates all user interaction with system. | Identify Phenomenon, MainMenu, Saved Images Page, Help, Settings |

| Class Name: MainMenu | |
|--|-----------------------------|
| Responsibility: | Collaborators: |
| Main graphical interface that user will identify phenomenon's attributes with. | User Interaction Controller |

| Class Name: Saved Images Page | |
|---|-----------------------------|
| Responsibility: | Collaborators: |
| Graphical interface where user will be able to see all images saved through this application and have the option to share it to their social media. | User Interaction Controller |

| Class Name: Help | |
|--|-----------------------------|
| Responsibility: | Collaborators: |
| Graphical interface where user will be able to see intructions on how to naviagate and use this application. | User Interaction Controller |

| Class Name: Settings | |
|--|-----------------------------|
| Responsibility: | Collaborators: |
| Graphical interface where user will be able to change the settings of the application. | User Interaction Controller |