SwimInGreece

Andrea Tsilogiannis – ISPW Project 2023/2024

1.Software requirement specification

1.1 Introduction

The aim of this document is to show the functionalities of the system and show how the phases of software development were approached in this scenario, and also how the SwimInGreece system was implemented.

1.2 Overview of the defined system

SwimInGreece is a platform which allows swimmers and sea lovers from all around the world to discover closely the beauty of the Greek sea, doing what they love in the meantime. There are two main user categories: swimmers, that can book trips, and organizers, which are likely locals sponsoring their homelands using an already existing service, which are tours around the islands by boat. Organizers can sponsor those tours by inserting them in the system, allowing the swimmers to see them.

1.3 Requirements

In order to run the software, the java runtime environment must be installed on the machine.

1.4 Related systems

At the moment there is no specific platform with the same use case as SwimInGreece, since it’s a local business and it’s not very famous online. There are anyway some similar systems, such as:

- TripAdvisor: only displays reviews and prices, doesn’t allow online booking for an entire trip;
- Yelp: as TripAdvisor it only displays generic reviews about many fields.

1.5 User Stories

1. As a swimmer, I want to book a swimming trip, so that i can plan my stay in advance.
2. As a swimmer, I want to attach reviews to a past trip, so that other users can decide on a trip plan.
3. As an organizer, I want to create a swimming trip experience, so that swimmers from other countries can visit new places.

1.6 Functional requirements

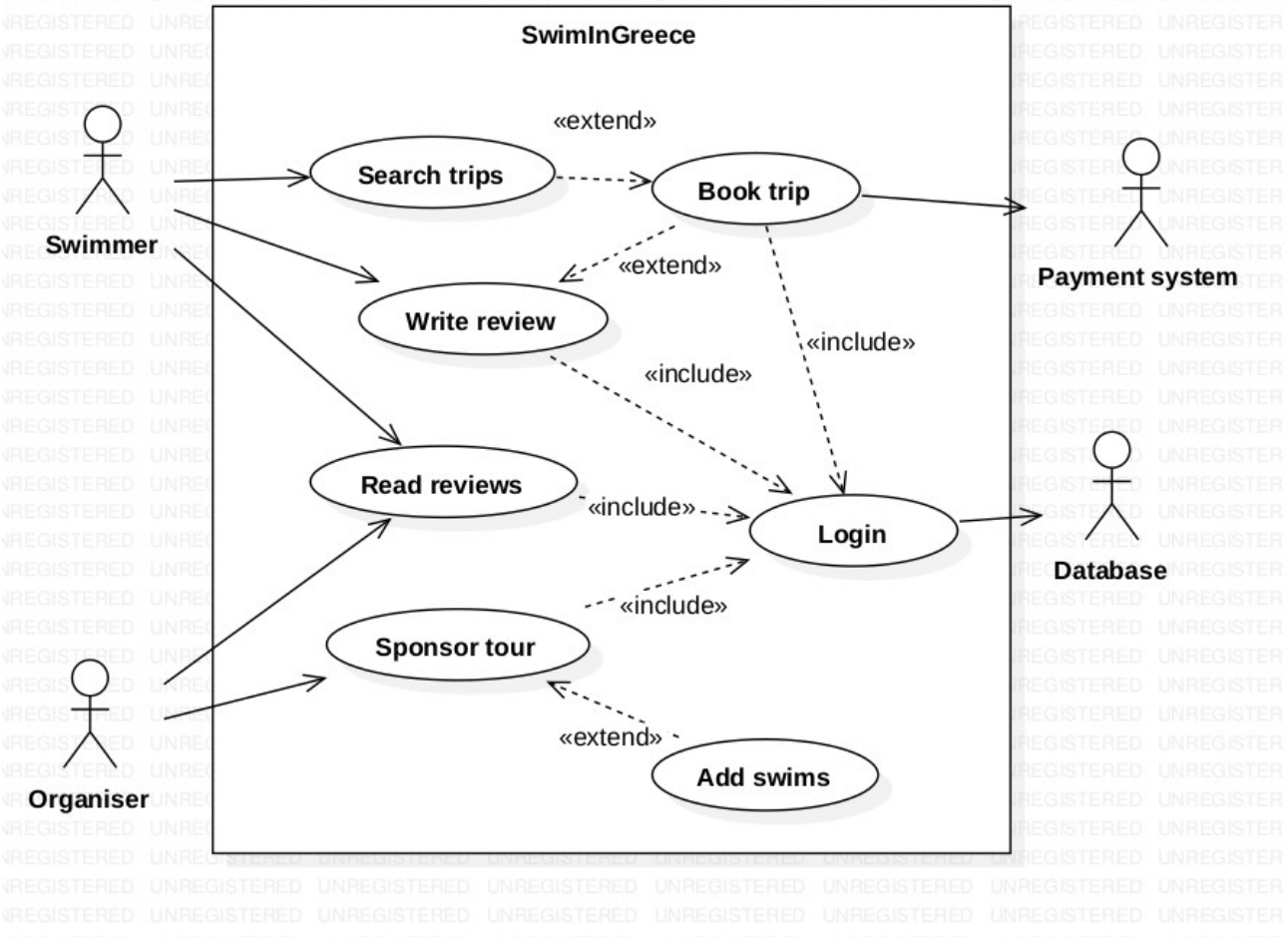
1. The system shall provide the tours for every island.
2. The system shall provide a review system.
3. The system must allow the adding of new tours.

Dictionary:

1. Tour: group of swims made in different days organized by an organizer.
2. Swim: single swimming event

1.7 Use case overview diagram

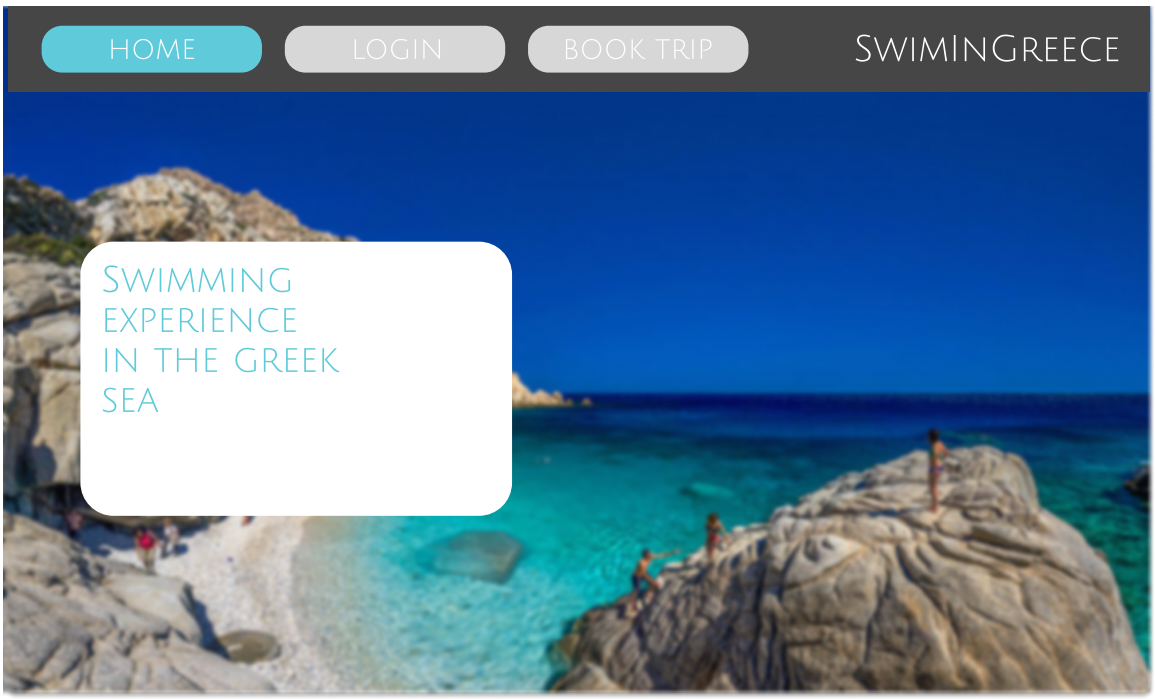
A pdf version is attached in order to have a closer look to it.

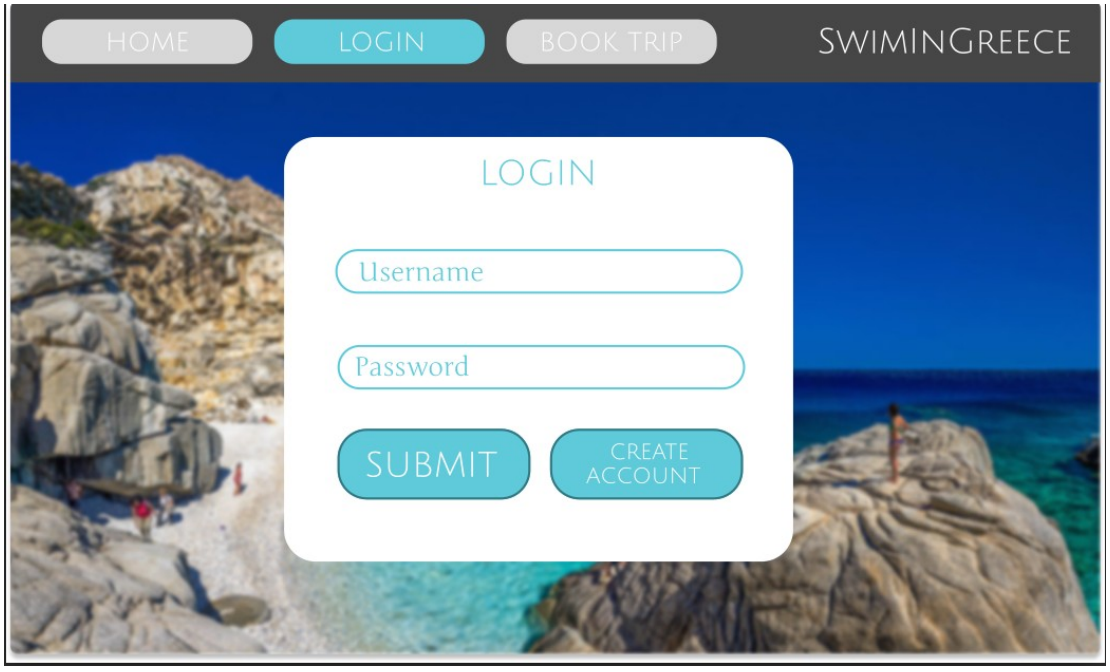


1.8 Internal steps
Name: Book tour
1. *Login
2. User clicks on Book tour button
3. System displays tour submission form
4. User inputs data
5. User clicks on “next step” button
6. System displays swims submission form
7. User inputs data and submits
8. System saves data in database
9. System displays homepage

Extensions:
5a/7a. Empty fields: system displays an error message
8a. Tour already exists: abort operation and display error message

2. Storyboards

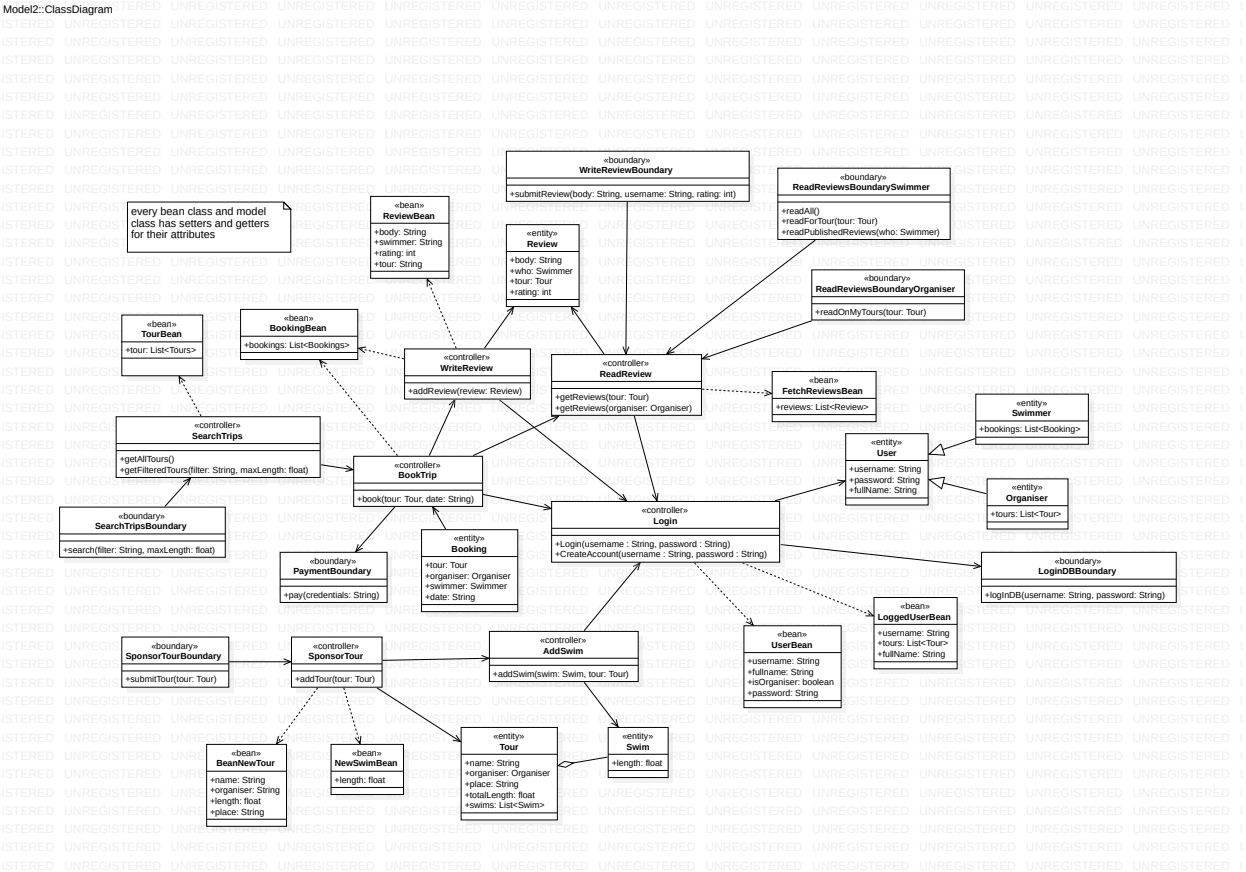




3. Design

3.1 VOPC

A pdf version of all the following diagrams are attached to this document.



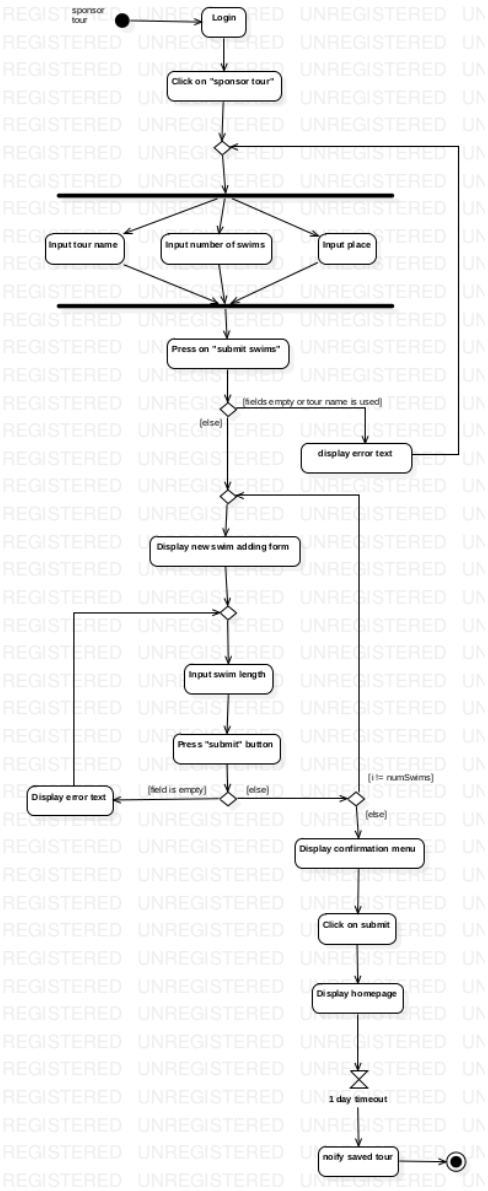
3.2 Design-level diagram

The diagram has many elements and can’t be displayed properly on this document.
Referencing the pdf file: MVC.pdf
The patterns used were:
- singleton pattern: used on the database connector, so that the dbms has to handle connections only from one source; used also on the Model class, in order to load faster the views when fxml files are swapped between each other;

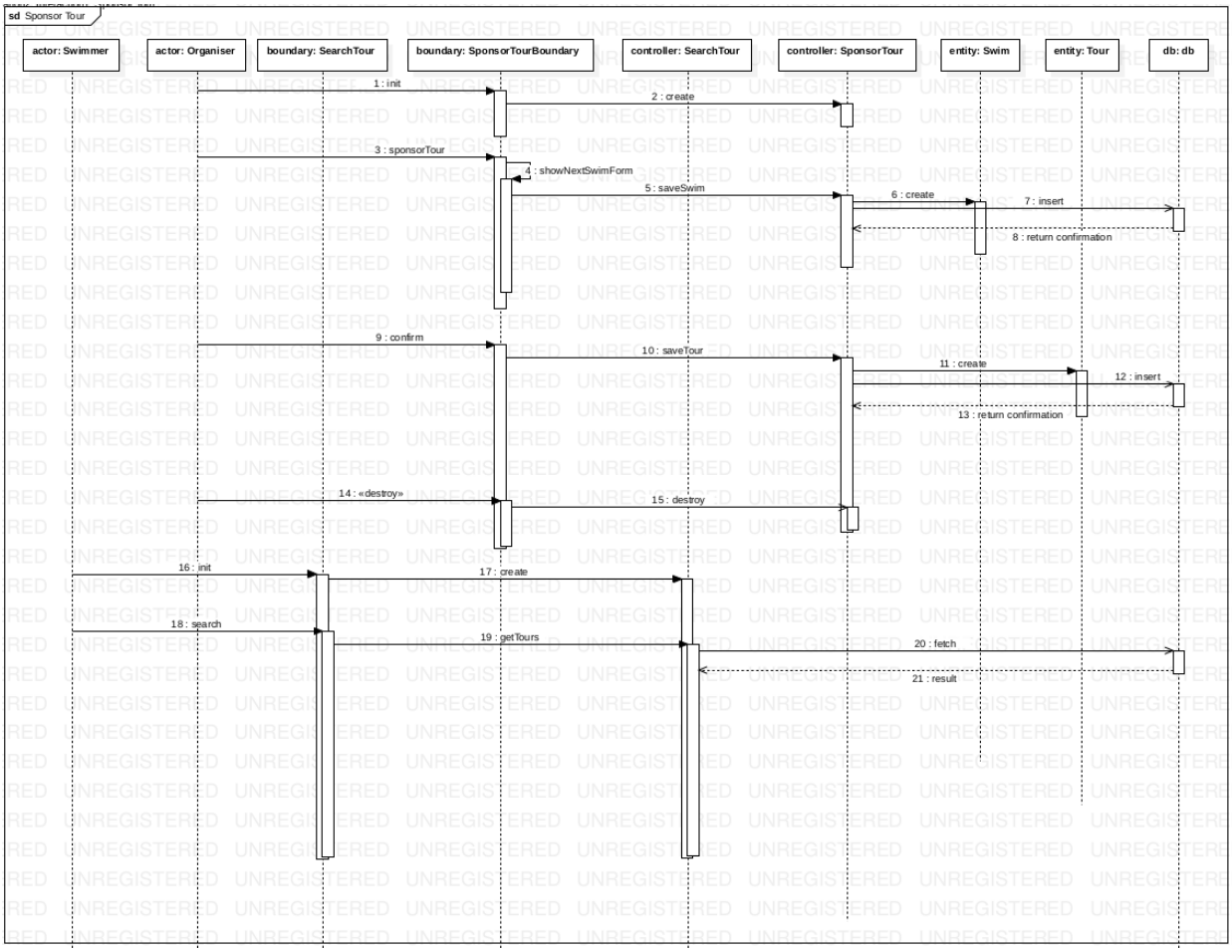
- facade pattern: used in the Facade class, that was used to recreate the instance of a User or a Tour with a single interface, in order to make this operation more easy and hide the implementation behind every Dao class and Query class to the calling class;

- factory method pattern: used in the ViewFactory class, used to create all the different views, create instances of the graphical controllers and bind those to the fxml files.

3.3 Activity diagram



3.4 Activity diagram



3.5 State diagram