Phase III: Software design and modeling.

System Objective

Property-Express is designed as a website that allows people to list, search, and rent short-term accommodations. Anyone visiting Albania for a short period of time (tourists, business people, etc.) can search for an apartment or house that is available for rent for a period of time ranging from 1 day to several weeks.

Modeling Objectives

Performance Criteria • Response time - interaction between the user and the maps are expected to be real-time. The user will place a marker on the map, displaying the requested address in real-time.

Dependency Criteria • Robustness - the system will respond immediately if user error occurs. On one hand, in cases where the user enters incorrect credentials or attempts to access a service without proper authorization, the system warns and stops the user from continuing further. • Availability - Property-Express will be available online after completion. • Security - the new integrated Google Maps system strengthens the security of the website and its users by ensuring that every address is legitimate and reflecting it on a specific map.

Maintenance Criteria • Readability - Currently, the system is easy to read due to the rigorous organization of files according to functionalities. • System change - Given that finding the code that describes a functionality is easy, it is easy to modify it. • Portability – Property-Express is accessible from any internet-connected electronic device. The Google Maps API provides map display on mobile devices.

End-User Criteria • Utility - For the reasons mentioned above regarding security and error prevention, we can say that the system will meet every user's needs and support their activities. • Usability - The graphical interface of the system is designed to be easy and usable for almost any target group: young, old, Albanian, and foreign.

Definitions, Acronyms, Abbreviations • We will refer to rental properties as "houses". • User - a logged-in website user who has the right to list and rent houses. • Visitor - an unlogged website user who only has the right to search and view listed houses. • Administrator - has all the rights of a regular user, but also has other rights related to viewing the website database, and extracting various statistics related to the website.

System Modelling

*Context Models:* Context models are used to describe the environment in which the house rent application operates. The context model for a house rent application includes the following elements:

Actors: The actors in the context model are the tenants, landlords, and property managers who will use the application.

Use Cases: The use cases in the context model describe the tasks that the actors will perform using the application, such as searching for properties, managing rental agreements, and processing payments.

System Boundary: The system boundary in the context model defines the boundary of the application and identifies the interfaces between the application and the actors.

*Interaction Models:* Interaction models describe the way in which the actors and the application interact. The interaction model for a house rent application includes the following elements:

User Interface: The user interface is the means by which the actors interact with the application.

• Create profile: A form where the user enters their personal information. Visible when there are no logged-in users.

• Edit profile: A form where the user changes their information. Visible only to logged-in users.

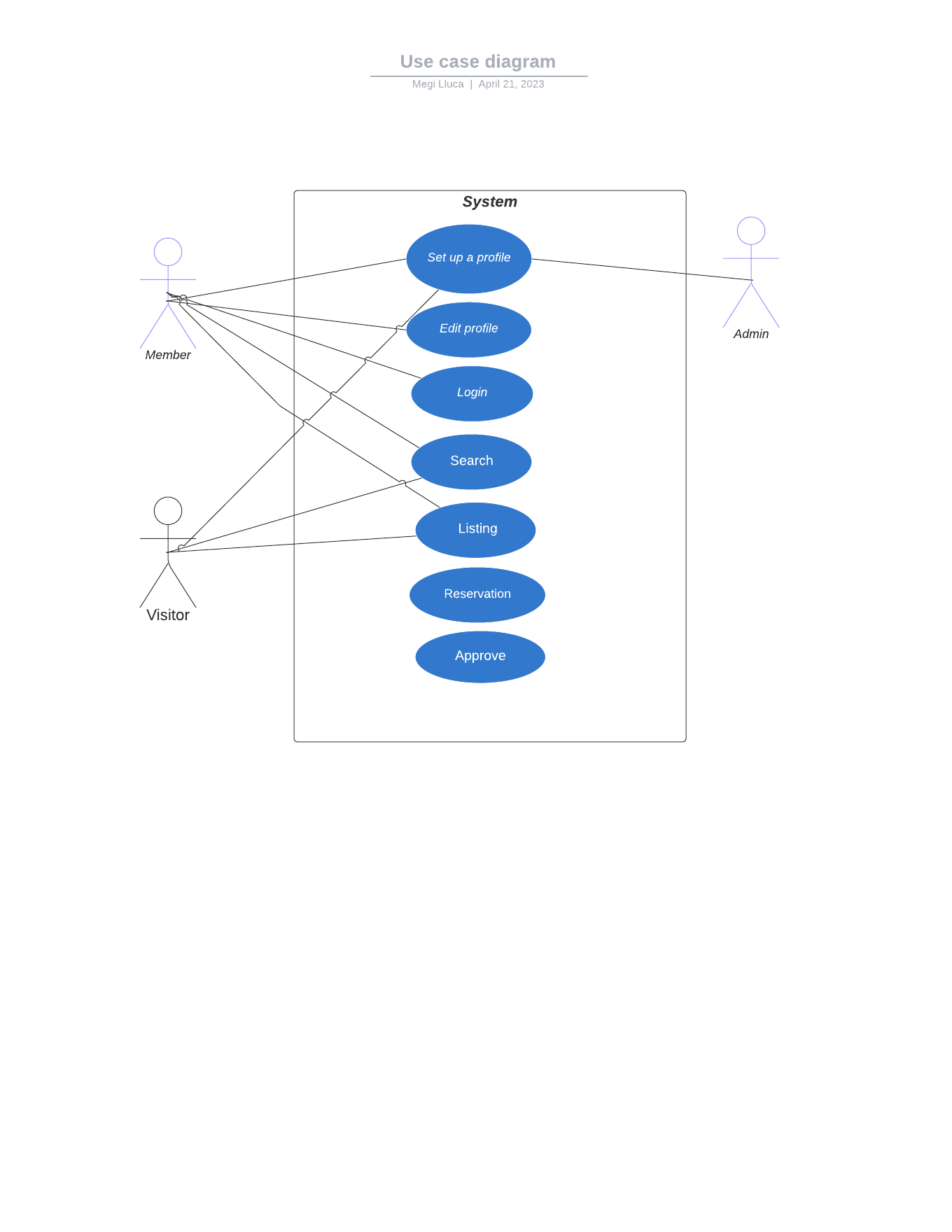
• Log in: The login interface.

• List home: A form through which the user enters data about their home. Visible only to logged-in users. • Edit home: Editing the features of a home. Visible only to logged-in users, on the profile of the logged-in user's home.

• Search home: The main form on the homepage. Visible to all types of users.

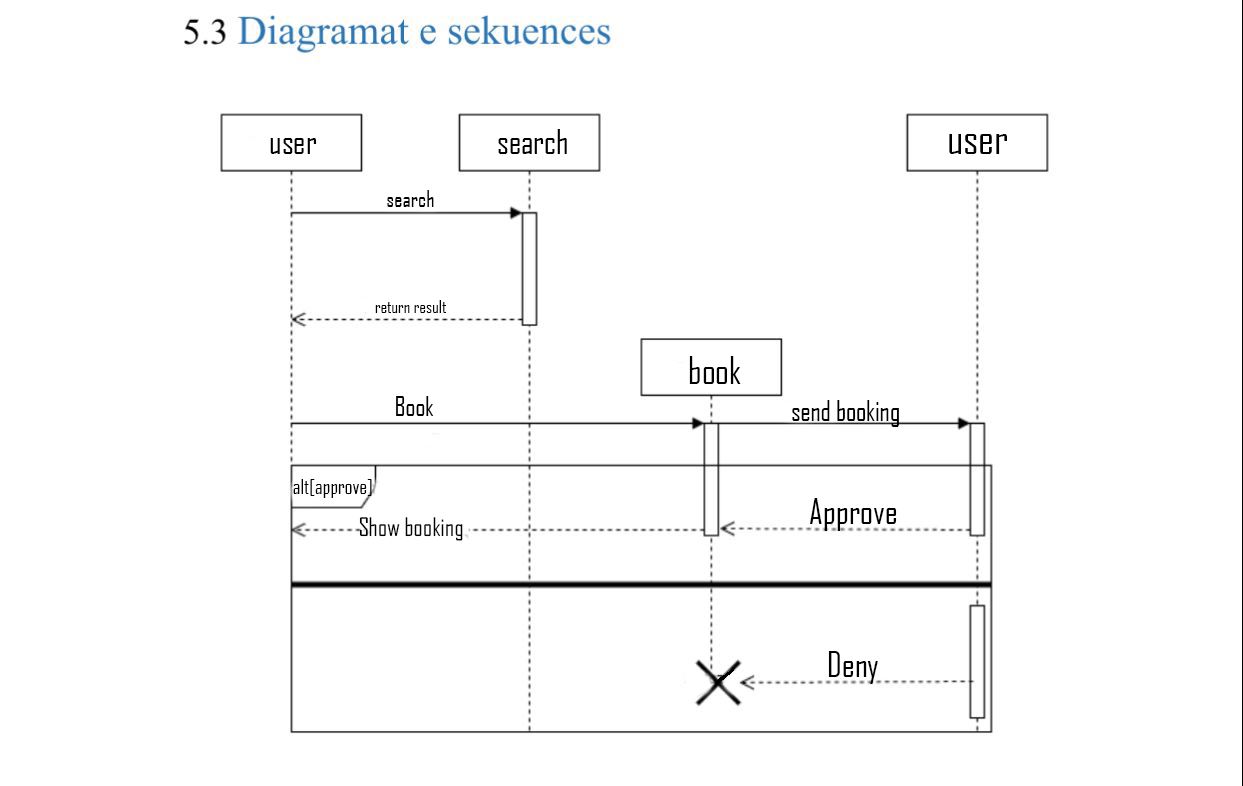
• Reserve: A button that enables the reservation of a home. Visible only to logged-in users, on any home profile.

• Approve reservation: A button that allows approval of a reservation request. Visible only to logged-in users.



Input and Output: The input and output of the application should be designed to provide feedback to the actors and to ensure that the data entered into the system is accurate.

Navigation: Navigation within the application should be simple and easy to follow to ensure that the actors can move through the application easily.



*Structural Models*: Structural models describe the internal structure of the application. The structural model for a house rent application includes the following elements:

Data Entities: The data entities in the structural model are the objects that the application will need to store, such as properties, tenants, landlords, rental agreements, and payments.

Relationships: The relationships between the data entities should be identified to ensure that the data is organized in a meaningful way.

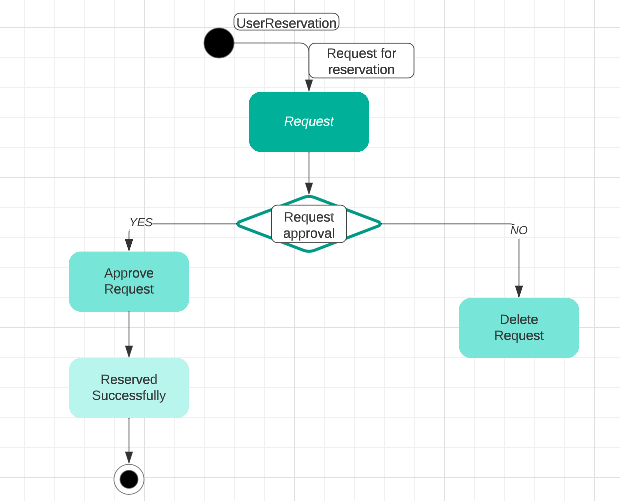
Database Schema: The database schema should be designed to support the storage and retrieval of the data entities and relationships.

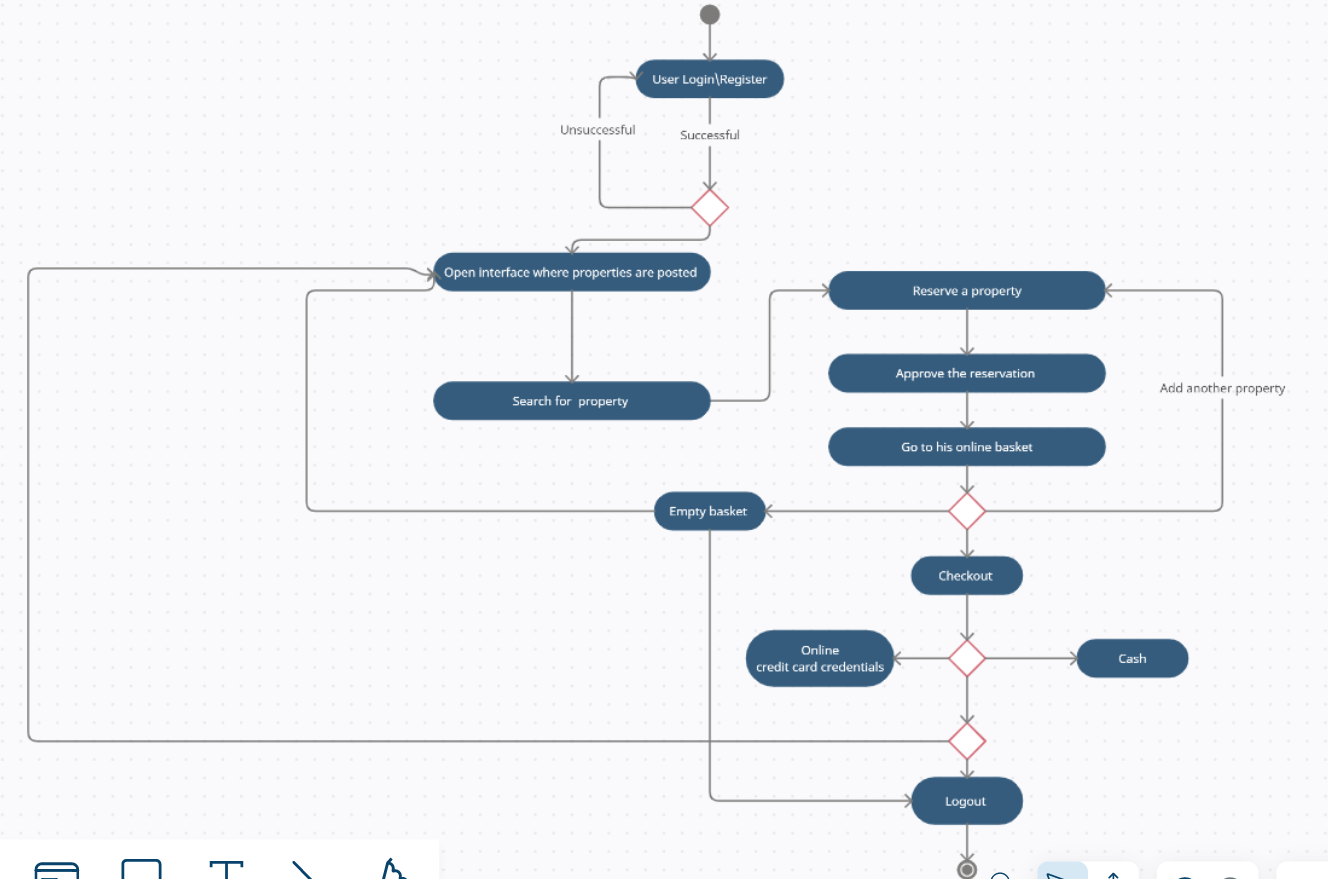
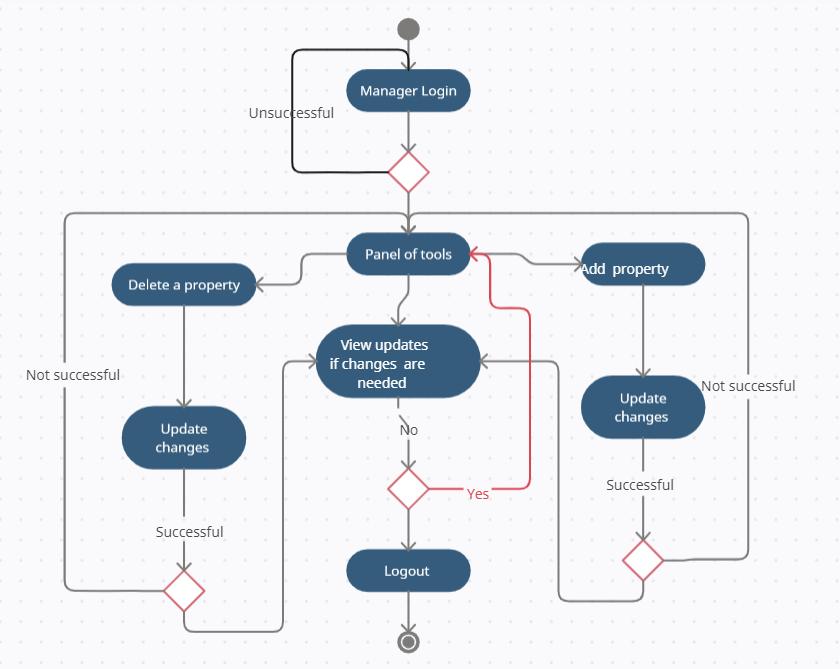


*Behavioral Models*: Behavioral models describe the behavior of the application in response to specific events or inputs. The behavioral model for a house rent application includes the following elements:

Use Case Scenarios: The use case scenarios describe the sequence of steps that the actors will take to perform a specific task using the application.

State Diagrams: State diagrams are used to model the behavior of the application in response to different inputs or events.



Activity Diagrams: Activity diagrams describe the flow of activities within the application, such as the process of creating a rental agreement.