

Polytechnic University of Puerto Rico
Electrical & Computer Engineering & Computer Science Department
Hato Rey, PR

Software Design Descriptions



Lambda Solutions Group Presents:



Name	Student ID	Program	Signature
Emilio Acosta Ortiz	86144	Computer Engineering	
Michael Quiles Melendez	83497	Computer Engineering	
Gabrielys Rivera Flores	89456	Computer Engineering	
Rafael Santiago Soliván	93818	Computer Engineering	
Manuel Seda Batista	120758	Computer Engineering	
Raúl Viruet Román	75747	Computer Engineering	

Changelog

Name	Date	Changes Performed	Version
Gabrielys Rivera	1/20/19	Began Elaboration of Section 4	1.00
Manuel Seda	1/20/19	Began L ^A T _E X template	1.01
Manuel Seda	1/20/19	Began working on Section 5	1.02
Raúl Viruet	1/21/19	Began working on Sections 3.3 & 6.2	1.03
Emilio Acosta	1/21/19	Began working on Sections 3.1 & 3.2	1.04
Rafael Santiago	1/21/19	Began working on Sections 6.1 & 6.2	1.05
Michael Quiles	1/22/19	Began working on Sections 1 & 2	1.06

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Overview	1
1.4	Definitions, Acronyms and Abbreviations	2
2	References	5
3	Decomposition Description	6
3.1	Module Decomposition	6
3.1.1	GUI Description	9
3.1.2	Database Description	10
3.1.3	Service description	10
3.2	Current Process decomposition	11
3.3	Data Decomposition	12
3.3.1	Graphical User Interface Decomposition	12
3.3.2	Service Decomposition	12
3.4	Database Decomposition	13
4	Dependency Description	14
4.1	Inter-Module Dependencies	14
4.2	Inter-Process Dependencies	14
4.3	Data Dependencies	15
5	Interface Description	16
5.1	Module Interface	16
5.1.1	Module 1 Detail: GUI	16
5.1.2	Module 2 Detail: Client	16
5.1.3	Module 3 Detail: Database	16
5.2	Process Interface	16
5.2.1	Process 1 Description: GUI	16
5.2.2	Process 2 Description: Client	17
5.2.3	Process 3 Description: Database	17
6	Detailed Design	18
6.1	Module Detailed Design	18
6.2	Data Detailed Design	18
6.2.1	Graphical User Interface Detailed Design	18
6.2.2	Database detailed Design	20
6.2.3	Service detailed Design	21

List of Figures

3.1	Module Decomposition Diagram	6
3.2	Class Diagram	7
3.3	Sequence Diagram	8
3.4	GUI	9
3.5	Database Description Diagram	10
3.6	Service Diagram	11
3.7	Current Process Decomposition Diagram	11
4.1	Modules Dependency Diagram	14
4.2	Process Dependency Diagram	14
4.3	Data Dependency Diagram	15
6.1	Database Entity - Relationship Diagram	21

List of Tables

1.1	Terms & Definitions	3
1.2	Terms & Acronyms	4
1.3	Terms & Abbreviations	4
3.1	GUI Decomposition Table	12
3.2	Service Decomposition Table	12
3.3	Database Decomposition Table	13
6.1	Administrative Account	18
6.2	User Account - Resident and Visitor	19
6.3	Users Table	20
6.4	Boat Table	20
6.5	Ticket Table	21
6.6	Required Data Validation - Administrative GUI	22
6.7	Required Data Validation - User GUI	22

1 Introduction

This section covers the purpose of the document, the scope of the project in regard of the design, an overview and some referential documents that will be used in the creations of this SDD.

1.1 Purpose

The purpose of this document Is to facilitate the analysis and the planning of the system “Ticket VC Island”. This document will help us in the development of the system as we go on. It will provide a detailed description of how the internal components of the system are related and how they communicate with each other. We will also include some diagrams explaining how the flow of the system works. This is the first version of this Software Design Description (SDD) which follows the standards of the Institute of Electrical and Electronic Engineers, 1963 (IEEE).

This document is a technical guide for the following Technical team members:

- Emilio Acosta Ortiz
- Rafael Santiago Sullivan
- Gabrielys Rivera Flores
- Michael Anthony Quiles
- Raul Viruet Roman
- Manuel Seda Batista

1.2 Scope

The touristic and residential boating system of Vieques and Culebra is a public transport system which takes residents and tourist from the island of Puerto Rico to the neighboring island of Vieques and Culebra. It uses high capacity boats on scheduled ride times on a daily basis. The central station is located in Ceiba, PR where one is able to buy tickets for a specific departure time to later board the ferry in said location. They have three scheduled times that roam around the times of 6:00 am, 10:30 am and 3:00 pm. The seats are unassigned, and they run on a first come first serve policy. The purpose of the system Ticket VC Island is to provide a better approach to the ticket distribution while also providing easier user interface. We seek to prioritize those who are residents of the islands to be able to get to their homes. Also, by utilizing computerized systems, we will be able to be more consistent in timings and numbered seating. Thanks to this we will grant the passengers speed and convenience in their purchases and help the administrators gain cost efficiency and reliability in the system.

1.3 Overview

The following topics will be discussed in this software design description:

1. Decomposition Description
 - 1.1. System Architecture
 - 1.2. Data Model

- 2. Dependencies
 - 2.1. Module Dependencies
 - 2.2. Process Dependencies
 - 2.3. Data Dependencies
- 3. Graphical User Interfaces
- 4. Activity Diagrams
- 5. Class Diagrams
- 6. State Diagrams
- 7. Sequence Diagrams
- 8. Package Diagrams

1.4 Definitions, Acronyms and Abbreviations

In this section definitions, acronyms, and abbreviations of important terms mentioned in the document will be provided. In Table 1.1 the definitions will be found, in Table 1.2 the acronyms, and in Table 1.3 the abbreviations.

Term	Definitions
completed transaction	A completed transaction is considered a transaction that underwent all necessary verification and integrity constraints. Furthermore data on the database must be successfully updated.
database	a collection of information that is organized so it can be easily accessed, managed and updated[1]
ectcetera	a number of other things or persons unspecified[2]
hardware	the physical aspect of computers, telecommunications, and other devices[3]
hash	a function that converts one value to another[4]
HTML	the standard markup language for creating Web pages[5]
IEEE	he world's largest technical professional organization dedicated to advancing technology for the benefit of humanity[6]
interface	a device or program enabling a user to communicate with a computer[7]
LSG	the name of the company that is proposing this system
MTA	a government-owned corporation of Puerto Rico charged with providing maritime transportation services for cargo and passengers within Puerto Rico and the island-municipalities of Vieques and Culebra[8]
QR code	a type of 2D bar code that is used to provide easy access to information through a smartphone[9]
query	a database query can be either a select query or an action query. A select query is a data retrieval query, while an action query asks for additional operations on the data, such as insertion, updating or deletion[10]
server	a computer that provides data to other computers[11]
software	a general term for the various kinds of programs used to operate computers and related devices[12]
TVCI	the name of the system proposed by LSG
URL	the address of a resource on the Internet; it indicates the location of a resource as well as the protocol used to access it[13]
Web page	a set of data or information which is designed to be viewed as part of a website[14]

Table 1.1: Terms & Definitions

Term	Acronym
HTML	Hyper Text Markup Language
IEEE	Institute of Electrical and Electronics Engineers
LSG	Lambda Solutions Group
MTA	Maritime Transport Authority
QR code	Quick Response code
SRS	Software Requirements Specifications [15]
TBA	To Be Announced
TBD	To Be Decided
TVCI	Tickets VC Islands
URL	Uniform Resource Locator

Table 1.2: Terms & Acronyms

Term	Abbreviation
etcetera	etc.

Table 1.3: Terms & Abbreviations

2 References

- [1] Margaret Rouse, Allan Leake and Adam Hughes, “Definition Database.” <https://searchsqlserver.techtarget.com/definition/database>. Online, Accessed: 2018-12-15.
- [2] Dictionary.com, “etcetera.” <https://www.dictionary.com/browse/etcetera>. Online, Accessed: 2018-12-15.
- [3] Margaret Rouse, “Definition Hardware.” <https://searchnetworking.techtarget.com/definition/hardware>. Online, Accessed: 2018-12-15.
- [4] TechTerms, “Hash Definition.” <https://techterms.com/definition/hash>. Online, Accessed: 2018-12-15.
- [5] w3schools, “Introduction to HTML.” https://www.w3schools.com/HTML/html_intro.asp. Online, Accessed: 2018-12-15.
- [6] IEEE, “What is IEEE?.” https://supportcenter.ieee.org/app/answers/detail/a_id/190/~/%3Fwhat-is-ieee%3F. Online, Accessed: 2018-12-15.
- [7] Oxford Dictionary, “Definition of interface in English.” <https://en.oxforddictionaries.com/definition/interface>. Online, Accessed: 2018-12-15.
- [8] Wikipedia contributors, “Puerto Rico Maritime Transport Authority.” https://en.wikipedia.org/wiki/Puerto_Rico_Maritime_Transport_Authority. Online, Accessed: 2018-12-15.
- [9] Margaret Rouse, “Definition QR code (Quick Response code).” <https://whatis.techtarget.com/definition/QR-code-quick-response-code>. Online, Accessed: 2018-12-15.
- [10] Margaret Rouse, “Definition Query.” <https://searchsqlserver.techtarget.com/definition/query>. Online, Accessed: 2018-12-15.
- [11] TechTerms, “Server Definition.” <https://techterms.com/definition/server>. Online, Accessed: 2018-12-15.
- [12] Margaret Rouse, “Definition Software.” <https://searchmicroservices.techtarget.com/definition/software>. Online, Accessed: 2018-12-15.
- [13] techopedia, “Uniform Resource Locator (URL).” <https://www.techopedia.com/definition/1352/uniform-resource-locator-url>. Online, Accessed: 2018-12-15.
- [14] Collins English Dictionary, “Definition of ‘web page’.” <https://www.collinsdictionary.com/dictionary/english/web-page>. Online, Accessed: 2018-12-15.
- [15] Inflectra, “What are System Requirements Specifications/Software (SRS)?.” <https://www.inflectra.com/ideas/topic/requirements-definition.aspx>. Online, Accessed: 2018-12-15.

3 Decomposition Description

This section of the SDD records the division of the software system into design entities. It describes the way the system is structured and the purpose and function of each entity. For each entity, it provides a reference to the detailed description. It uses the identification, type, purpose, function and subordinate's attributes.

3.1 Module Decomposition

"Tickets VC Island" has 3 different components, each one has different functions within the system. These different components are Graphical User Interface, Database and Service. As you can see in the following figure.

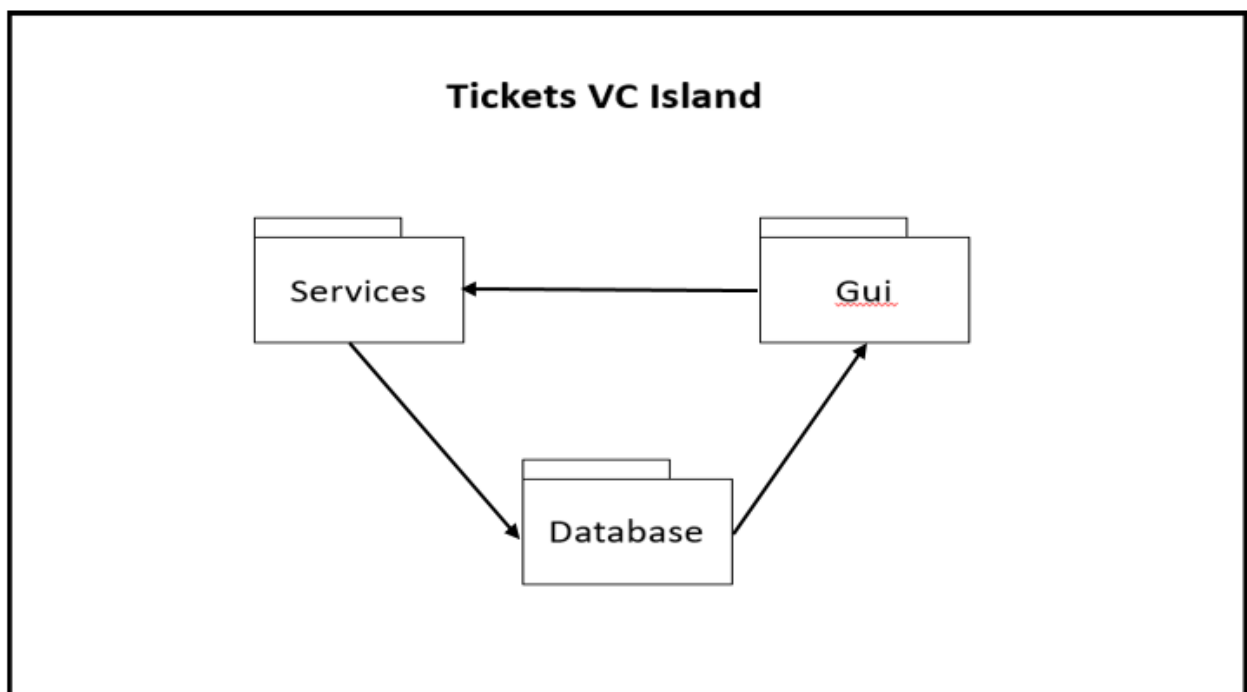


Figure 3.1: Module Decomposition Diagram

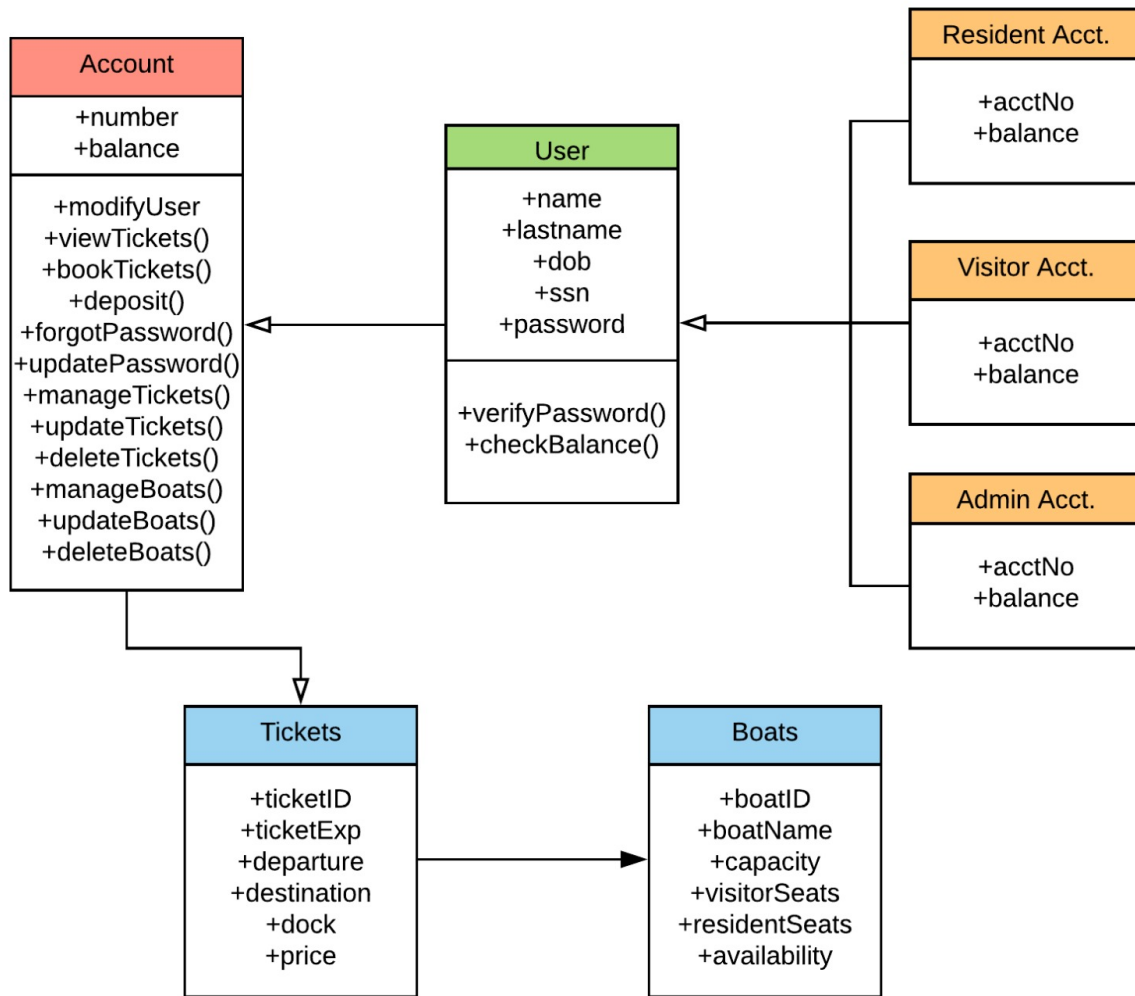


Figure 3.2: Class Diagram

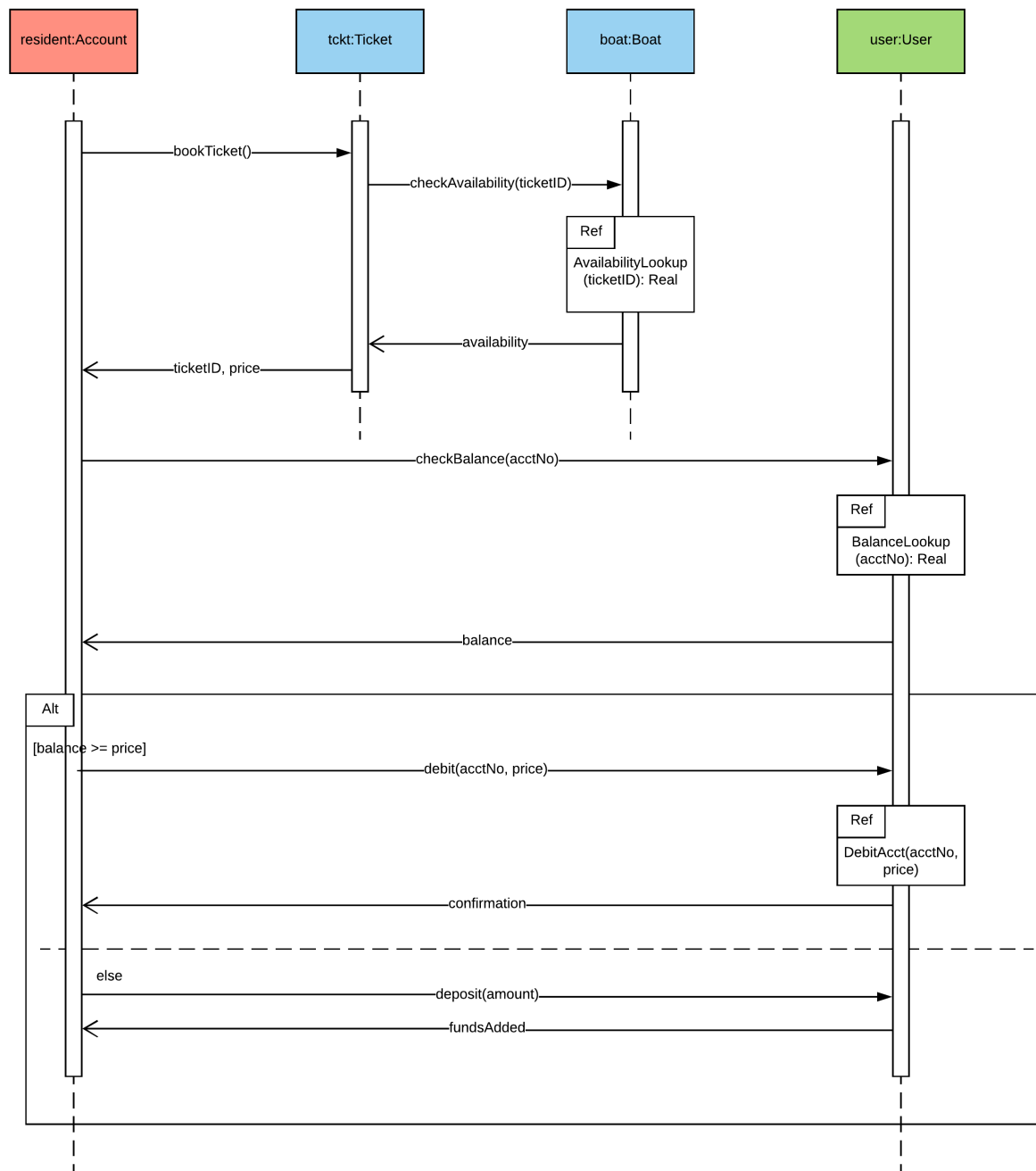


Figure 3.3: Sequence Diagram

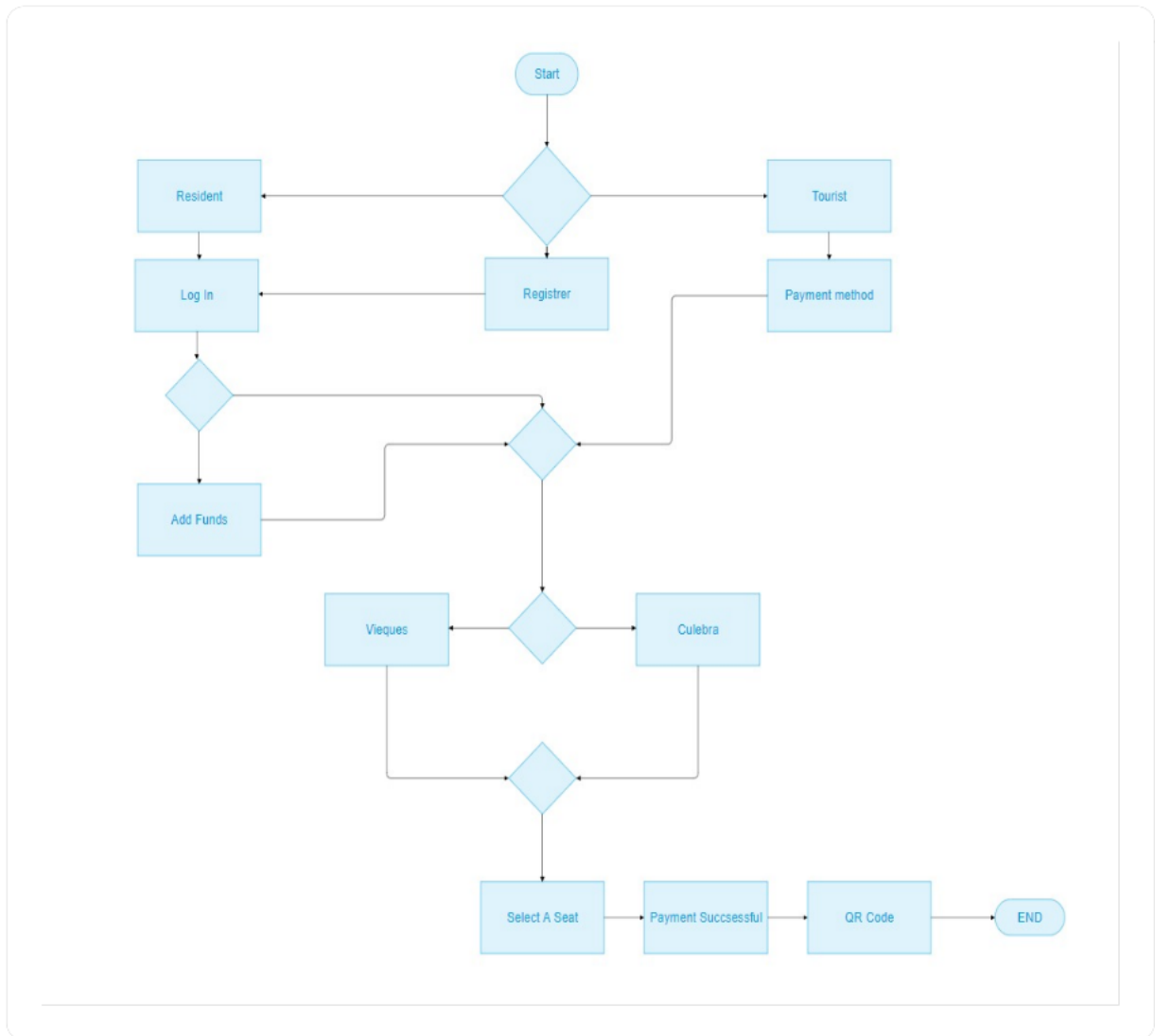


Figure 3.4: Activity Diagram

3.1.1 GUI Description

GUI: graphical user interface that has the database and interaction with the components of the Service. It is an essential part of the system, since the user interacts with it through the user interface.

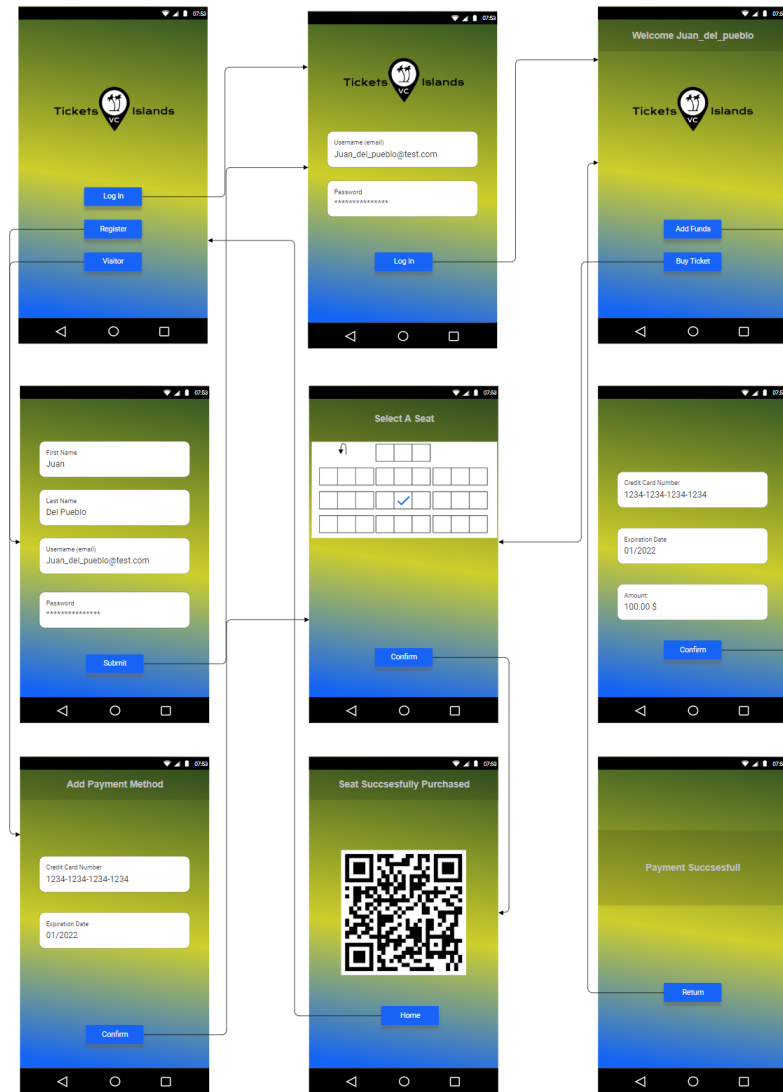


Figure 3.5: GUI

3.1.2 Database Description

Database: it is an external component, the stores and the amount of tickets purchased. In addition, the registration of all ticket purchases and the number of tickets available. Each domain object is stored securely in the database and is retrieved when necessary.

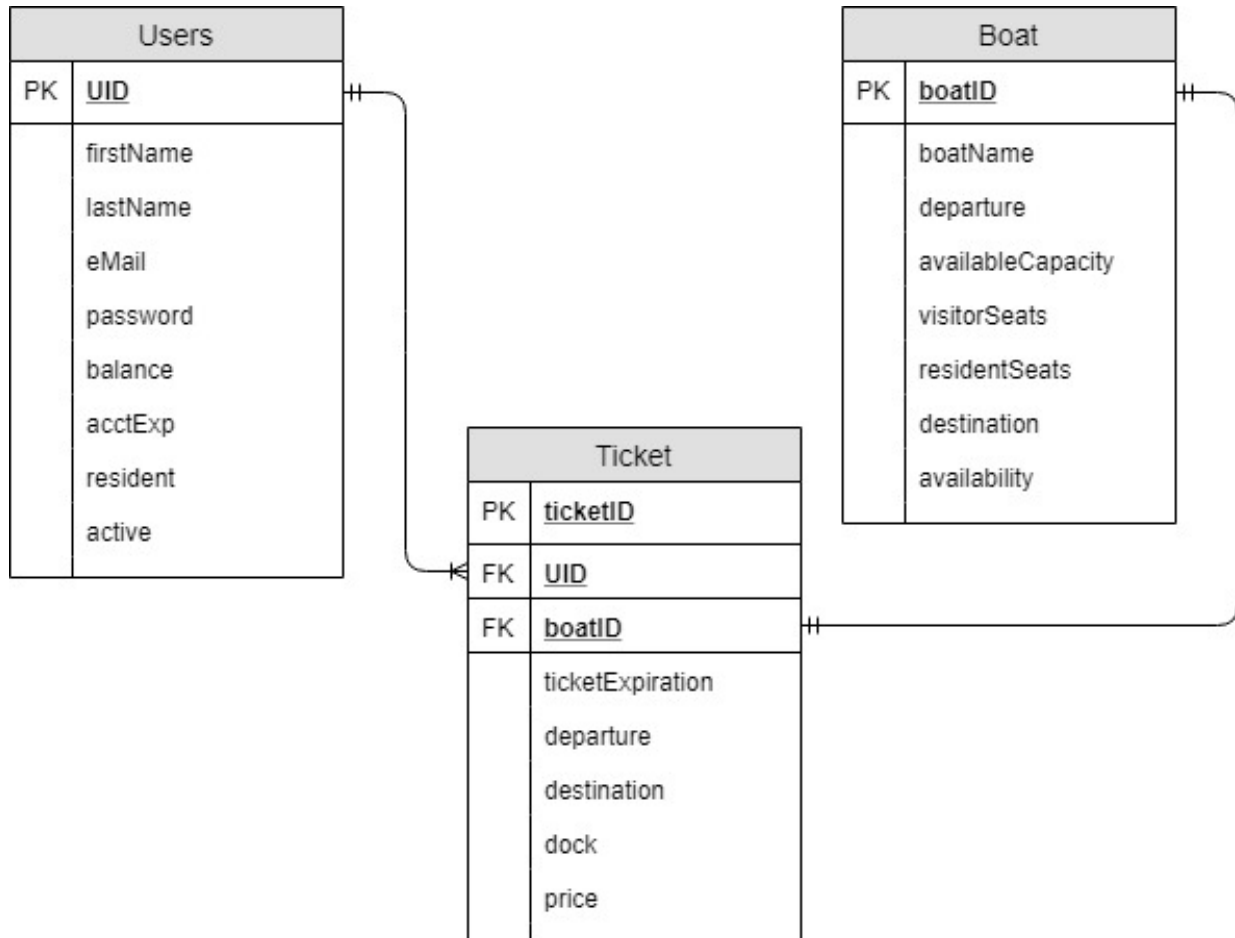


Figure 3.6: Database Description Diagram

3.1.3 Service description

This component includes the entities that are responsible for the logic. For example, the security service that controls login operations, among other entities, are elements of this component. All the work related to logic is done by the entities of this component.

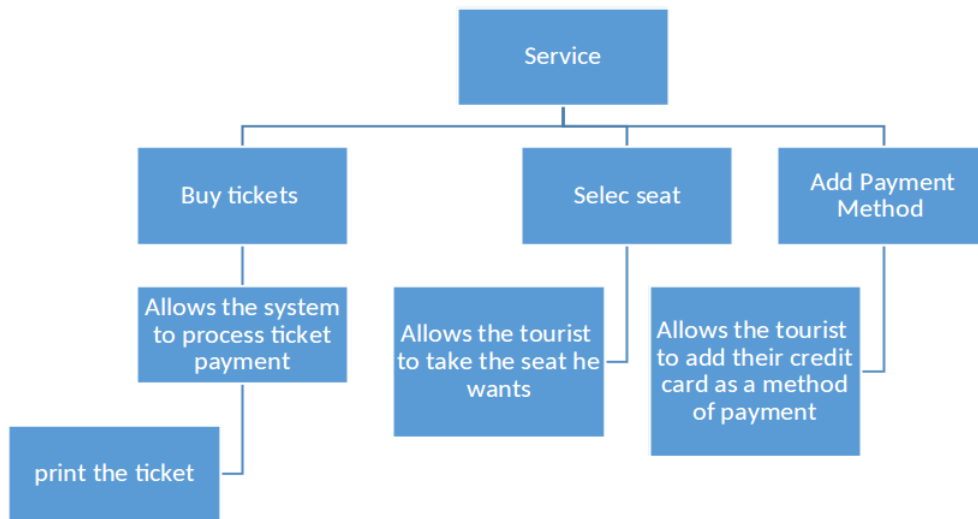


Figure 3.7: Service Diagram

3.2 Current Process decomposition

The system of "Tickets VC Island" will contain the following current processes decomposition.

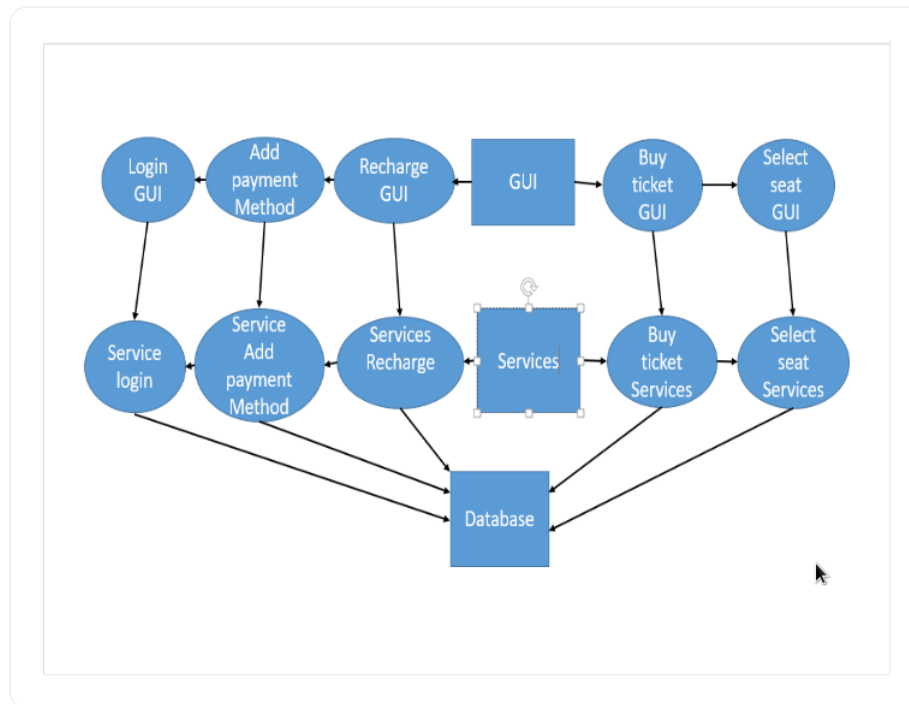


Figure 3.8: Current Process Decomposition Diagram

3.3 Data Decomposition

In this section there will be a brief description of how the system data is separated by entities.

3.3.1 Graphical User Interface Decomposition

Identification	GUI
Type	Package
Purpose	Enables user interaction with the system
Function	The GUI (Graphical User Interface) enables the user to interact with the system. Via this GUI the user will be able to purchase tickets, recharge account balance (For resident users only)
Subordinates	N/A

Table 3.1: GUI Decomposition Table

3.3.2 Service Decomposition

Identification	Service
Type	Package
Purpose	Component that undertakes logical procedures and interconnects database data with that's displayed on the GUI
Function	Data validation, logical operations, handling and dispatching request from database and GUI
Subordinates	Database, GUI

Table 3.2: Service Decomposition Table

3.4 Database Decomposition

Identification	Database
Type	Package
Purpose	Entities and their attributes are stored in the Database, Relationship between entities are handled in the Database as well. The system will access this database to retrieve user information
Function	Stores user data about balance, available boats and available seats for both resident and visitor usewrs. Data stored in the database is utilized for the sale of tickets to guarantee resident users have prioritized seating
Subordinates	N/A

Table 3.3: Database Decomposition Table

4 Dependency Description

This section of the SDD document describes the dependency relationships between different modules, processes and data of TVCI.

4.1 Inter-Module Dependencies

All the activities carried out in the TVCI web application are handled by the DB module. The operations that users request in the GUI module are managed by the service module. That is, the GUI module depends on the service module, which in turn, depends on the DB module, as can be seen below in Figure 4.1. To see each of the modules in more detail, please refer to section 3.1.

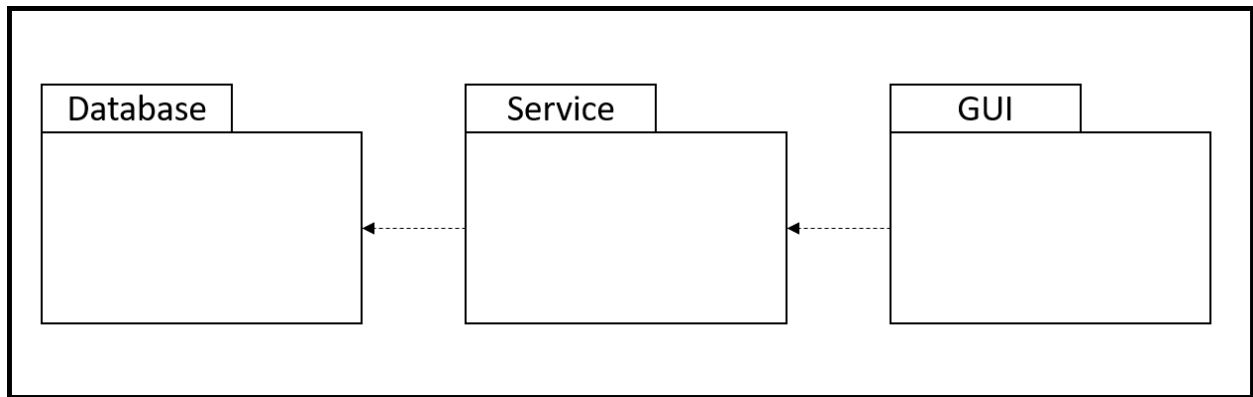


Figure 4.1: Modules Dependency Diagram

4.2 Inter-Process Dependencies

Inter-Process Dependencies are generally derived from business rules and express the order in which certain activities should be performed. All processes depend on users creating an account and then accessing it to make the purchase of tickets as shown in Figure 4.2. To see more in detail each of the processes, please refer to section 3.2.

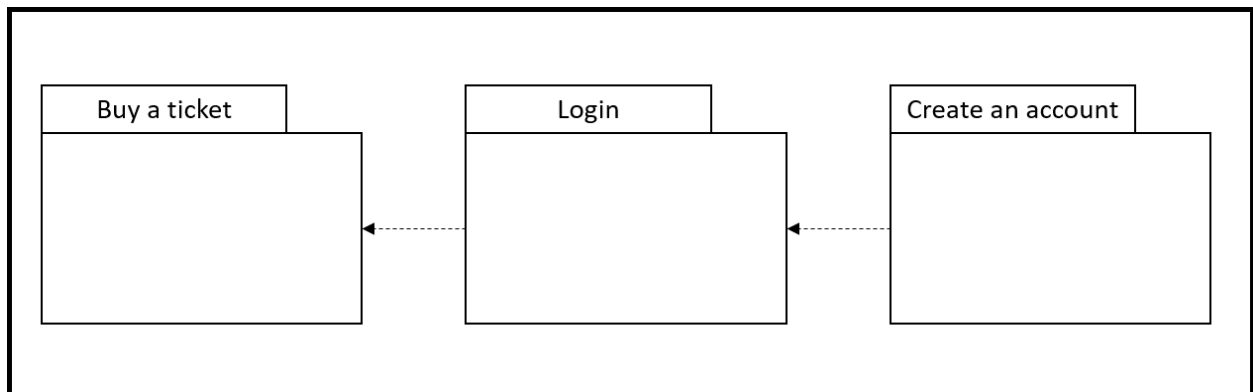


Figure 4.2: Process Dependency Diagram

4.3 Data Dependencies

Figure 4.3 describes the data dependencies of the TVCI service.

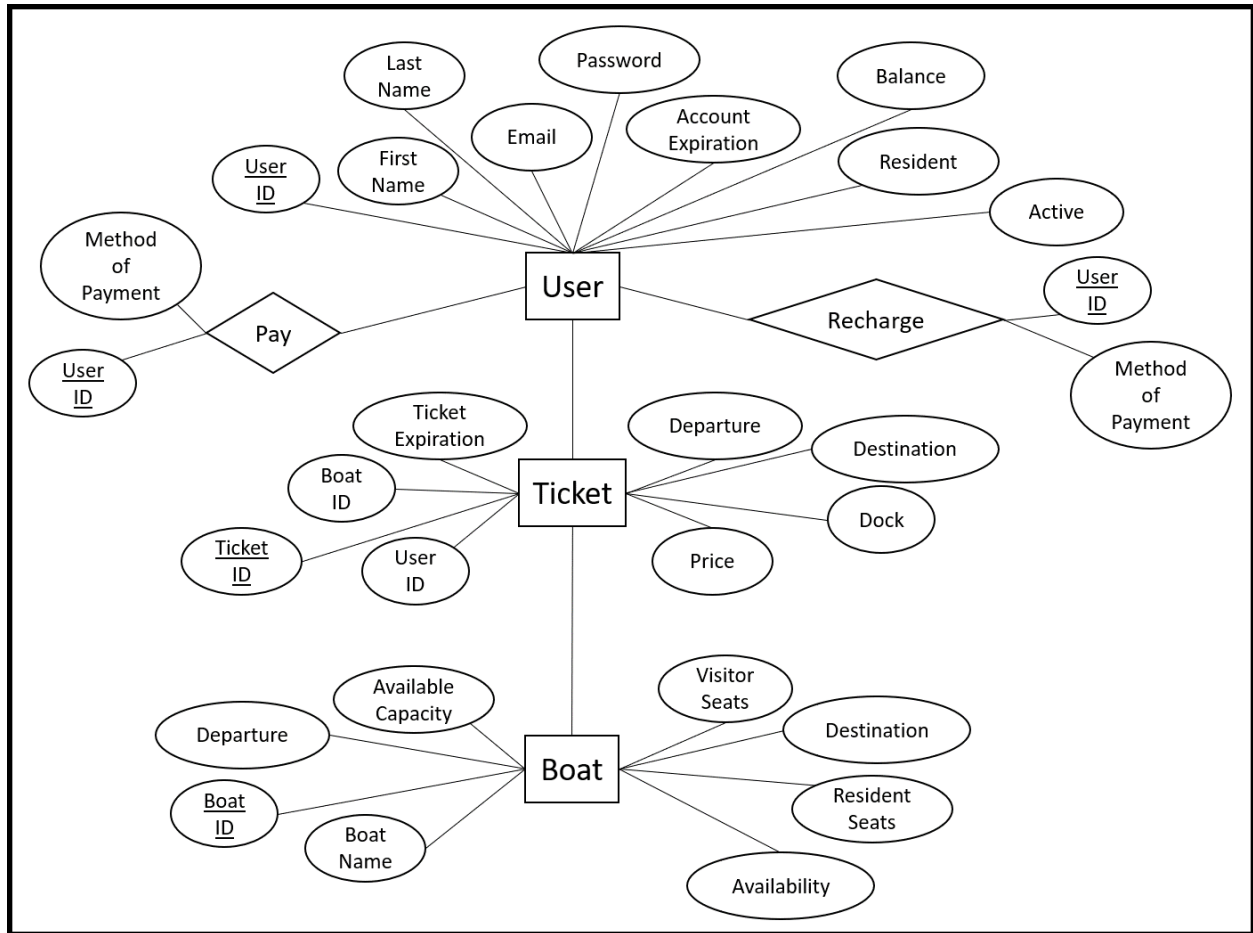


Figure 4.3: Data Dependency Diagram

5 Interface Description

This section establishes the expected interactions and behaviors of our product. Testing of the system must be based on the following sections.

5.1 Module Interface

In this section, we discuss how modules interact with each other and through which mechanism these interactions happen.

5.1.1 Module 1 Detail: GUI

The topmost module of the system. The user interacts with the program through this GUI, therefore, all user created changes and interactions are captured through this module. This module's functionality is directly subordinated to the client's functionality. The GUI is merely a graphical representation of the operations we intend to permit the user to control. Thus, the GUI's interactions are limited merely to calling the client's relevant functions. On its own, this module neither produces or changes data directly. All of its operations are executed through the client.

5.1.2 Module 2 Detail: Client

The client is the backbone of our system. All of the system's operations are coordinated by the client. The GUI's functionality calls client functions to produce the expected behavior. The client is also responsible of executing database queries or updates. Therefore, any action expected from our system is started in the client program.

5.1.3 Module 3 Detail: Database

This module organizes and serves most of the data our system needs to work. Since it's made in a SQL database, all of the interactions with the database are executed using SQL-queries. The client's purpose is to contain the queries relevant to all our system's interactions with the database. Therefore, if the system needs to get data from the database or modify data in the database, the SQL queries are defined and executed in the client. This module has no direct interactions with the GUI. Every interaction from within the system involving this module is only expected to happen in tandem with actions from the client. For maintenance and security purposes, the database will also interface with its DBMS, but in the perspective of our system behavior, this interaction will have no bearing in the definitions of the expected behaviors.

5.2 Process Interface

This section clarifies the internal rules that define each module's behavior. Any behavior clashing with these rules results either in an amendment to the rules if intended and desirable, or the finding of a bug if unintended and undesired.

5.2.1 Process 1 Description: GUI

The GUI's internal rules are simple: no function contained in this module produces any data on its own. All user provided data is simply handed over to the client for further processing. The views displayed by the GUI depend on the user interacting with the system. An MTA clerk will interact with the clerk GUI, a resident will interact with the resident ticket purchasing GUI and the tourist will interact with the generic

ticket selling GUI. Both clerks and residents have to log-in to use the system; tourists can complete their transactions without entering any personal credentials other than their payment method.

5.2.2 Process 2 Description: Client

The client's internal rules are the most extensive ones. All GUI functions and all database transactions are handled by the client. Therefore, the user provided data gathered by the GUI is sent to the database through this module. This module may produce data for the GUI only when it is based on data found in the database. In other words, the only data produced by the client is data that is originated in the database, but processed in the client for other used (ie. price calculations). The client produces no original data.

5.2.3 Process 3 Description: Database

All user-provided data ends up here. MTA's data also ends up here. The only way for the system to interact with the database is through SQL statements that are handled by the client. The database will interact with the user through views, so the users can't access any data not relevant to their operations.

6 Detailed Design

6.1 Module Detailed Design

As part of a design organization, all entities and details of the software we are designing is going to be describe in this section. Starting with a section giving a brief description of how the software is going to be divided by entities, which are three. The first is a Graphical User Interface Design, giving a detailed description of this design as well as the functions performed in this entity. The second entity is the Database Design, where a detailed description of what the database is based on and the data type used in each table. The third and final entity is the Service Design, which detailed the processes used in the system, like retrieving and validating data, as well as displaying data in the GUI. With the of organization of these three entities we can have a better understanding of the importance for this system.

6.2 Data Detailed Design

6.2.1 Graphical User Interface Detailed Design

Detailed view of how the GUI operates and the data a user may enter to it to be processed further.

Item	Description
Main Menu	Prompts whether to add a resident to the system or edit a current resident's information
Administrative Log In	Needed to access administrative features. Allows to continue with log in procedure using the provided credentials or cancel the operation to return to the home menu
Administrative Log In Failure	Alerts user their log in was unsuccessful, prompts the user if they want to retry log in or cancel the operation
Resident Registry Menu	Allows to add data about a new resident user, or cancelling this operation to return to the home menu
Resident Registry - Review Information	Shows the temporary Resident information so it can be reviewed by the resident user and accept/decline said information. [Pertains to editing user information and new resident creation]
Search Resident Menu	Allows to search a resident by ID, by registered email address, or by first and last name
Select Resident Menu	Shows results of the search operation. Allows to choose a particular Resident Account to be edited or canceling the Resident edit operation
Resident Edit Menu	Allows to edit data about a Resident user, or cancelling this operation to return to the home menu

Table 6.1: Administrative Account

Item	Description
Language Screen	Allows User to choose preferred language for user interface
Main Menu	Users can choose which action they would like to perform, or return to the language selection menu
Purchase Menu	User Can choose to purchase tickets as a visitor or as a resident
Resident Log In	Allows a resident user to log in to the system using it's credentials
Resident Log in Failure	Alerts user their log in was unsuccessful, prompts the user if they want to retry log in or cancel the operation
Ticket Purchase Menu - Resident User	Allows a resident user to choose the destination where they would like to travel to, as well as the departure date and time for the corresponding ship. Allows the user to proceed with a transaction and purchase a ticket matching their specifications or canceling transaction to return to the previous menu
Ticket Purchase menu - Redient User - Lack of funds screen	Alerts the user that purchase was unsuccessful due to lack of funds and prompts the user if they would like to add funds to their account or cancel the transaction
Timeout Error Screen	Alerts the user that the transaction could not be completed due to a timeout error. Returns the user to the main screen
Ticket Purchase menu - Transaction completed screen	Alerts the user that the transaction was completed successfully, and prompts the user to collect their newly printed ticket
Ticket Purchase Menu - Visitor User	Allows a visitor user to choose the destination where they would like to travel to, as well as the departure date and time for the corresponding ship. Prompts the user to choose their payment method or cancelling transaction to return to the previous menu
Payment Method Screen	Allows users to enter their credit card information to make a payment, or cancel the operation.
Added Funds successfully screen	Notifies the Resident user their funds were added successfully and displays their current available balance
Help Screen	Provides the user a guide of how to use the GUI

Table 6.2: User Account - Resident and Visitor

6.2.2 Database detailed Design

A database is used to store and access user information. Data which has been entered via the GUI and validated on the service will be stored on the database. Likewise data from the database can be accessed and be displayed on the GUI.

Datatypes and Entities used in the Database

Attribute	Data Type	Constraints
UID	Integer	PK, NN
firstName	VARCHAR(20)	NN
lastName	VARCHAR(20)	NN
eMail	VARCHAR(50)	
password	VARCHAR(50)	NN
balance	DOUBLE	NN
acctExp	TIMESTAMP	NN
active	BOOLEAN	NN
resident	BOOLEAN	NN
destination	VARCHAR(10)	

Table 6.3: Users Table

Attribute	Data Type	Constraints
boatId	INTEGER	PK, NN
boatName	VARCHAR(20)	
departure	TIMESTAMP	
availableCapacity	INTEGER	
visitorSeats	INTEGER	
residentSeats	INTEGER	
destination	VARCHAR(10)	
availability	BOOLEAN	NN

Table 6.4: Boat Table

Attribute	Data Type	Constraints
ticketId	VARCHAR(20)	PK, NN
UID	INTEGER	NN
departure	VARCHAR(10)	NN
destination	VARCHAR(10)	NN
boatId	INTEGER	NN
ticketExpiration	TIMESTAMP	NN
ticketPrice	DOUBLE	NN

Table 6.5: Ticket Table

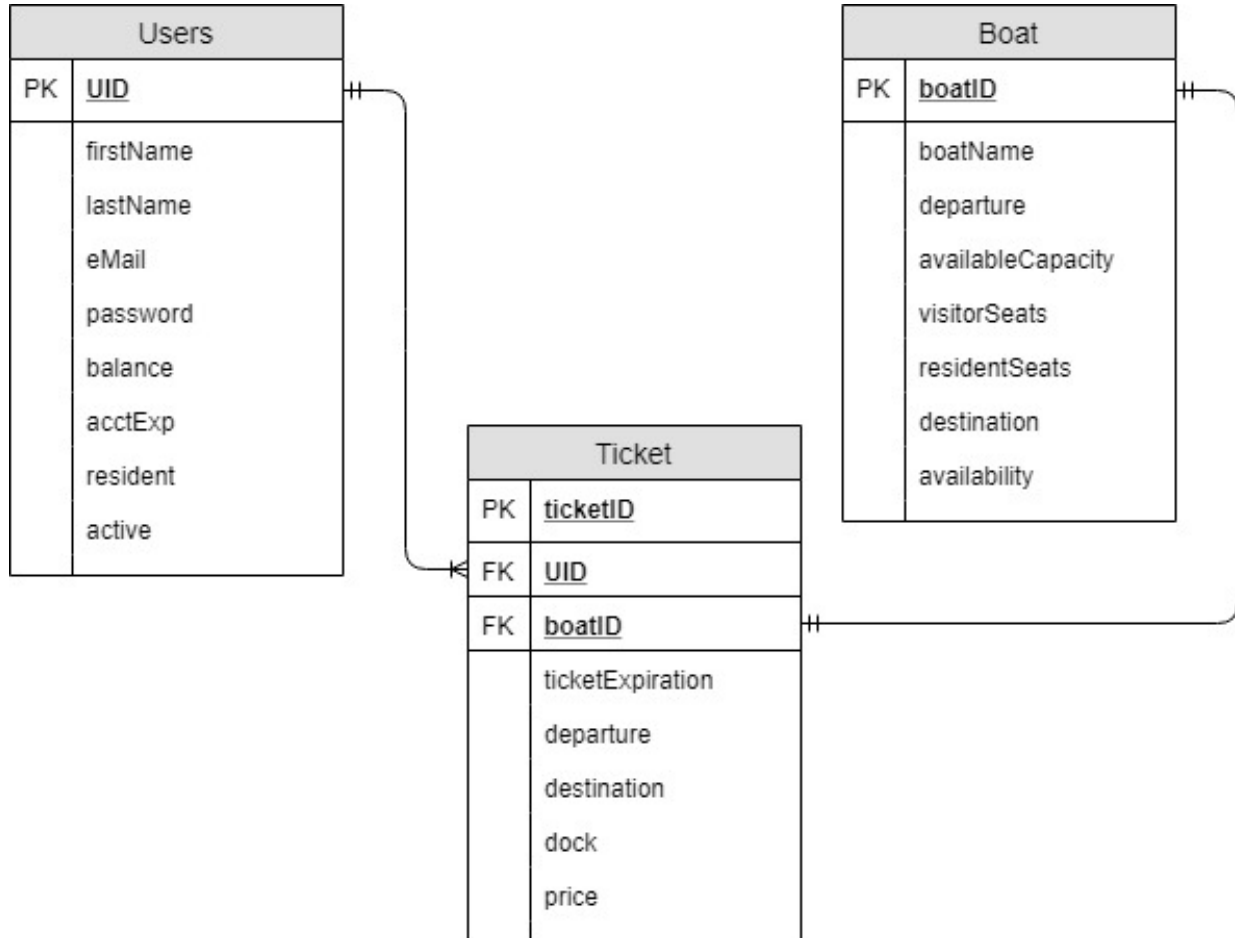


Figure 6.1: Database Entity - Relationship Diagram

6.2.3 Service detailed Design

The service is the functional aspect of the web based system. it's in charged of data validation for data entered in the GUI before storing said data in the database, as well as fetching data from the database to be displayed in the GUI.

Data Element	Constraint
Username	Matches an username for administratice account
Password	Matches a password for a particullar administrative account. Should also be not null and have a minimum of 6 characters
First Name	Should not be Null
Last name	Should not be null
Password - Resident	Should not be null and have a minimum of 6 characters
Confirm Password - Resident	Should be exactly identical to the password provided in the "Password" field, be not null, and have a minimum of 6 characters
e-Mail	Should be in the format 'local-part@domain'
User ID - Search Field	Should not be Null and be composed of numerical integer vaues
New Password	Should not be null and have a minimum of 6 characters

Table 6.6: Required Data Validation - Administrative GUI

Data Element	Constraint
e-Mail	Should be in the format 'local-part@domain' and match an email address for a registered Resident user
Password	Matches a password for a particullar Resident account. Should also be not null and have a minimum of 6 characters
Ticket Purchasing information	Matches information about actual ship departure and arrival times to the selected destination. Also displays the correct price for said travel option
Available Balance	Refelects the accurate available balance associated with a particular Resident user account
Payment method	Validates that credit card information is valid using the Luhn Algorithm

Table 6.7: Required Data Validation - User GUI