Cairo University  
Faculty of Computers and Information



**CS251**

**Software Engineering I**

GOFO

Software Design Specifications

Contents

[Team 3](#_Toc40625000)

[Document Purpose and Audience 3](#_Toc40625001)

[System Models 4](#_Toc40625002)

[I. Class Diagram(s) 4](#_Toc40625003)

[II. Class Descriptions 5](#_Toc40625004)

[III. Sequence diagrams 5](#_Toc40625005)

[Class - Sequence Usage Table 8](#_Toc40625006)

[IV. User Interface Design 9](#_Toc40625007)

[Tools 9](#_Toc40625008)

[Ownership Report 9](#_Toc40625009)

# 

# Team

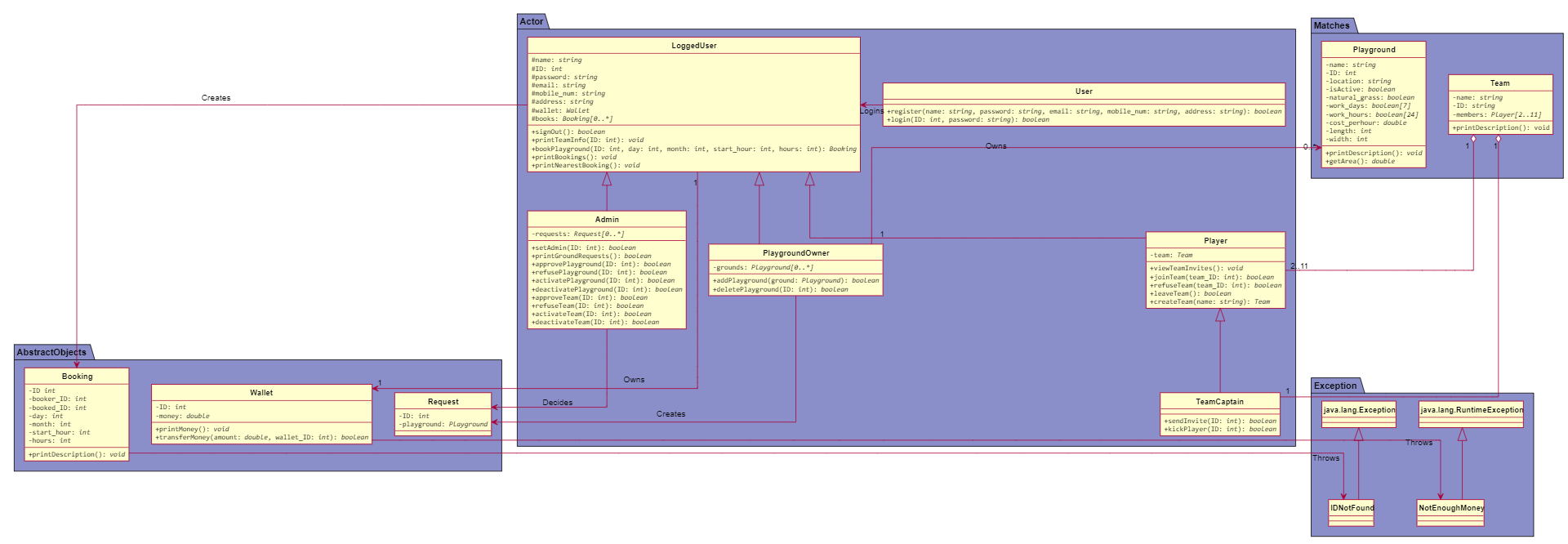
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20180250 | Mahmoud Ashraf | [ma5027300@gmail.com](mailto:ma5027300@gmail.com) | 01102488789 |
| 20180304 | Nader Fikry | [naderfikry245@gmail.com](mailto:naderfikry245@gmail.com) | 01097632795 |
| 20180193 | Fady Emad | [fadyemad14705@gmail.com](mailto:fadyemad14705@gmail.com) | 01203732443 |

# Document Purpose and Audience

* This program helps players to find and book football fields without wasting much time and effort.
* This document describes the shape, structure and design of the program through models.
* There are two main audiences in the program first, the player who wants to book football fields second, is the playground owner who wants to provide playgrounds fields.

# System Models

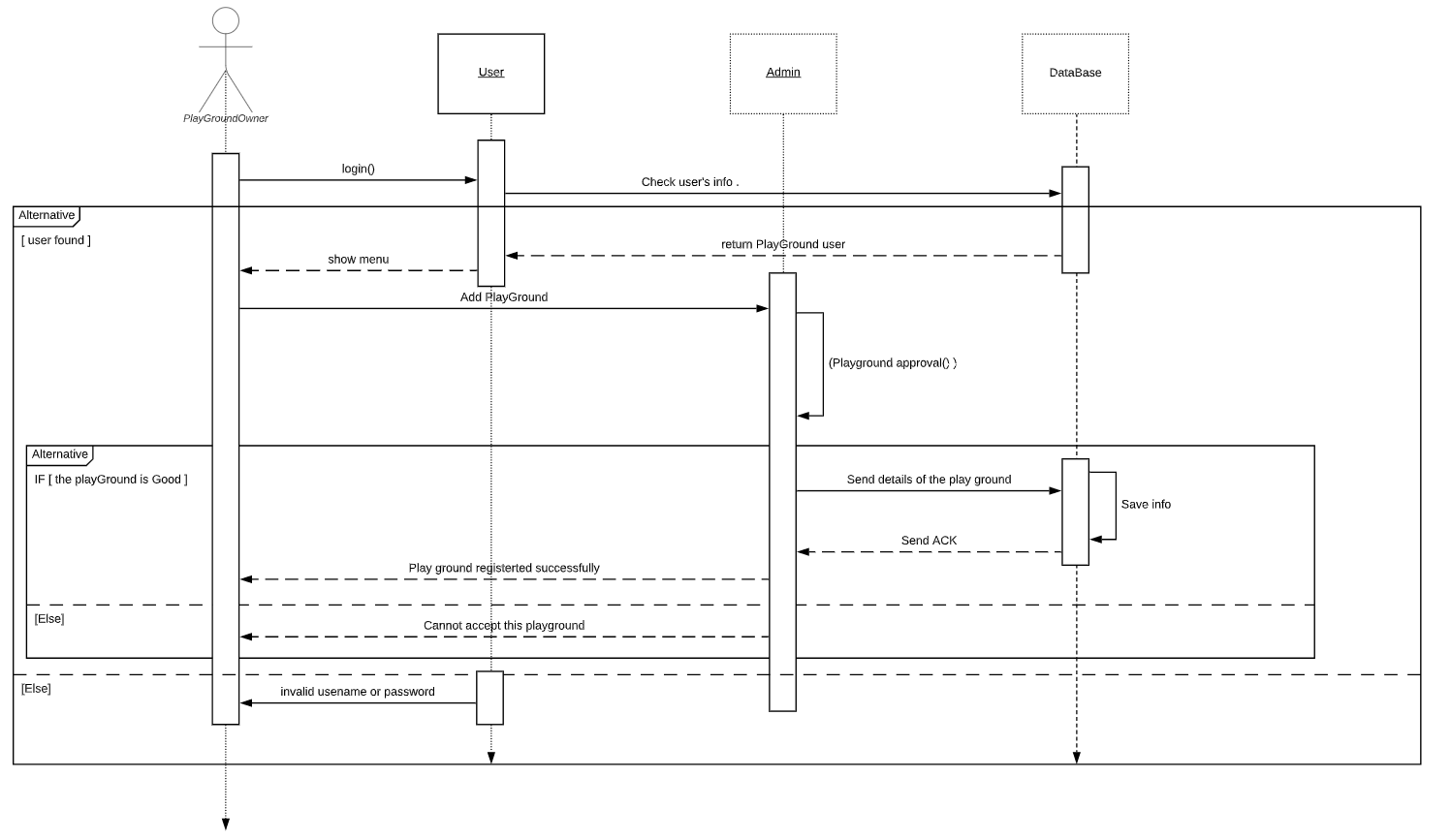
## Class Diagram(s)

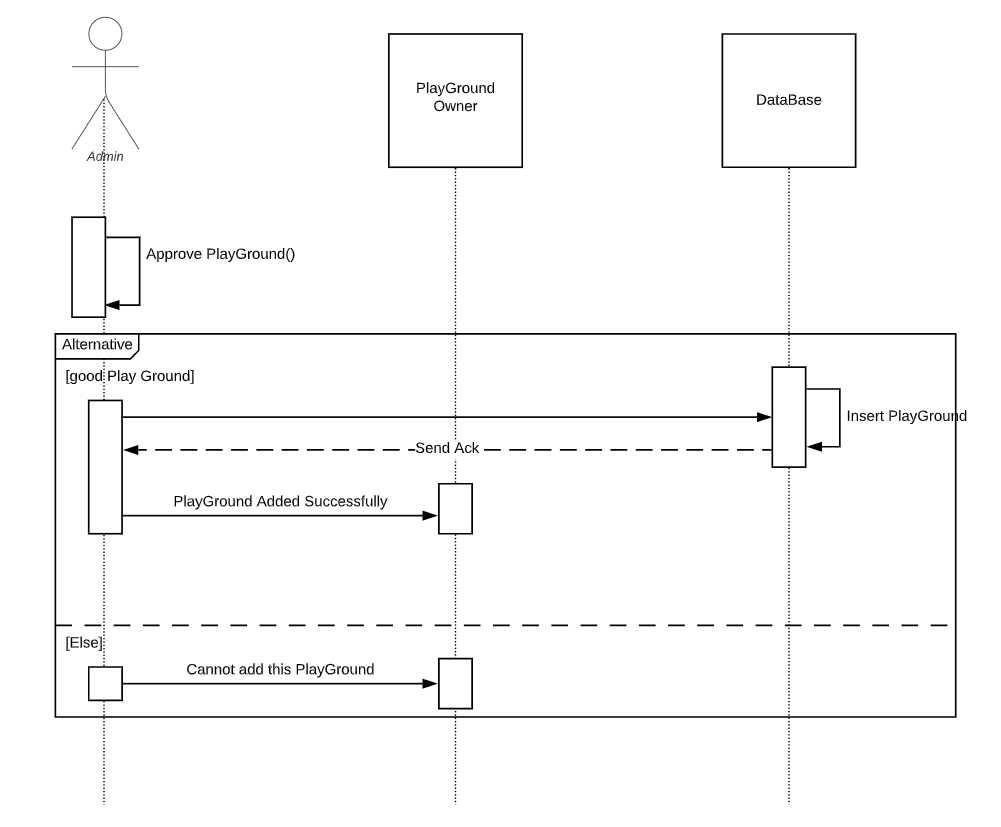


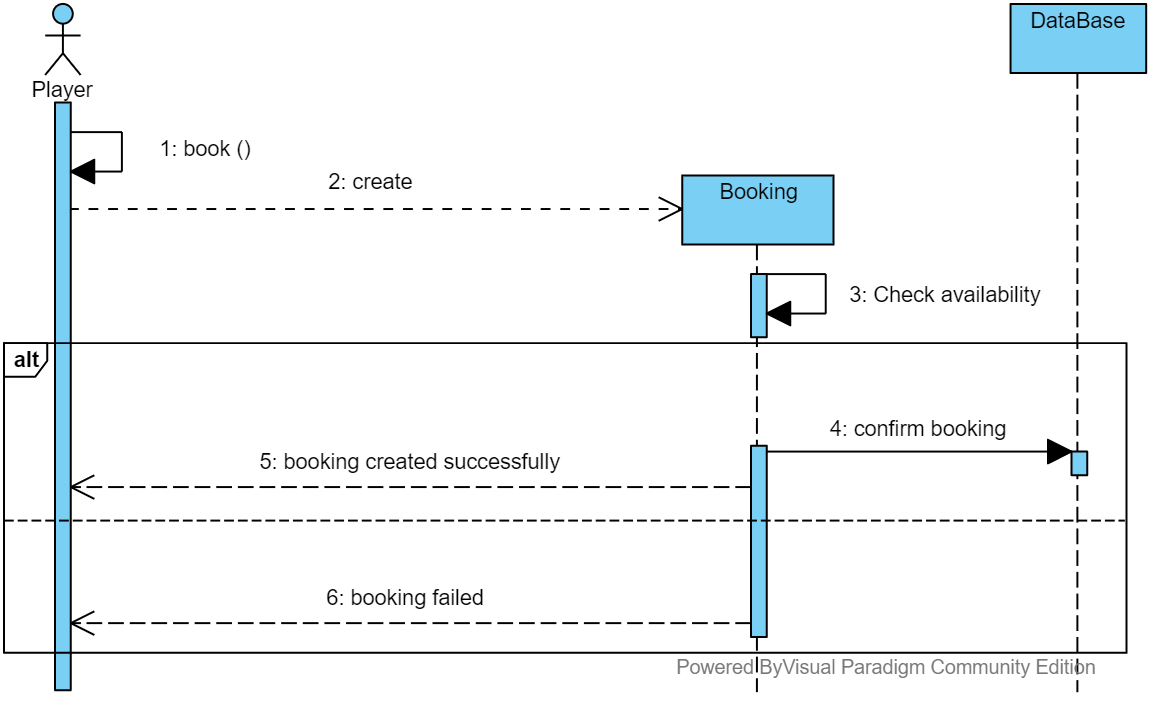
## Class Descriptions

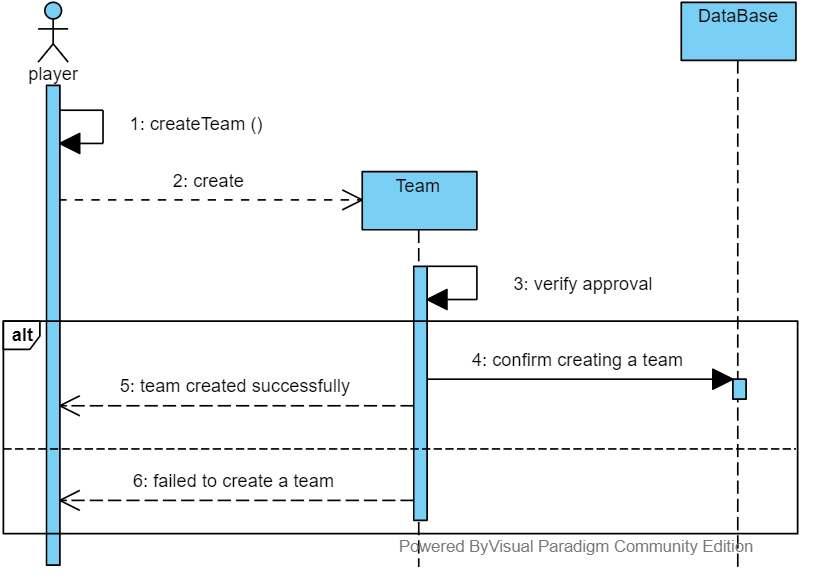
| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1. | User | A class that represents a user that is neither registered nor logged in. A user that has just opened a program and only has two responsibilities: to login or register. |
| 2. | LoggedUser | A class that represents a user that is registered and is logged in. Classes: Player, PlaygroundOwner and Admin inherit from this class. Responsibilities: Sign out, Book a playground, view his bookings and check his wallet. |
| 3. | Player | A class that represents a player. Responsibilities: All LoggedUser responsibilites + create a team, view Teams he is invited to and accept or refuse invitations.(See TeamCaptain for more info) |
| 4. | PlaygroundOwner | A class that represents a playground owner, who owns 1 or more playgrounds. Responsibilities: Add or Remove owned Playgrounds. An admin must accept or refuse such requests. |
| 5. | Admin | A class that represents a user with special privileges. An admin is responsible for managing the program. He is responsible for Managing Playground and Team requests. And suspending Playgrounds with bad history. |
| 6. | TeamCaptain | A class that inherits from a player. It represents a normal player but who has created his own team. A team must have one and only one TeamCaptain He is responsible for sending invites to other players and kick players from his team. |
| 7. | Playground | A class that represents a playground. It has many important attributes to determine its name, its location, if it is active, if it has natural grass, its area, its work days, its work hours and its cost. |
| 8. | Team | A class that is created when a player creates his own team. It has only one captain and up to 10 normal players. It also has a name. |
| 9. | Booking | A class to represent a booking that is made by a player to a playground. It saves the ID of the booker, the ID of the booked playground, total cost and the date and duration of the booking. |
| 10. | Wallet | A class that represents a wallet that saves a LoggedUser’s money. Depositing money to the wallet happens out of the system’s boundary. |
| 11. | Request | A class that represents a request made to the admins. It could be a Playground request or a Team creation request. It is deleted when responded by one of the admins. |
| 12. | IDNotFound | An exception that is thrown whenever the user specifies an ID of a player, playground.. etc that doesn’t exist. |
| 13. | NotEnoughMoney | An exception that is thrown when trying to book a playground but the wallet doesn’t have enough money. |

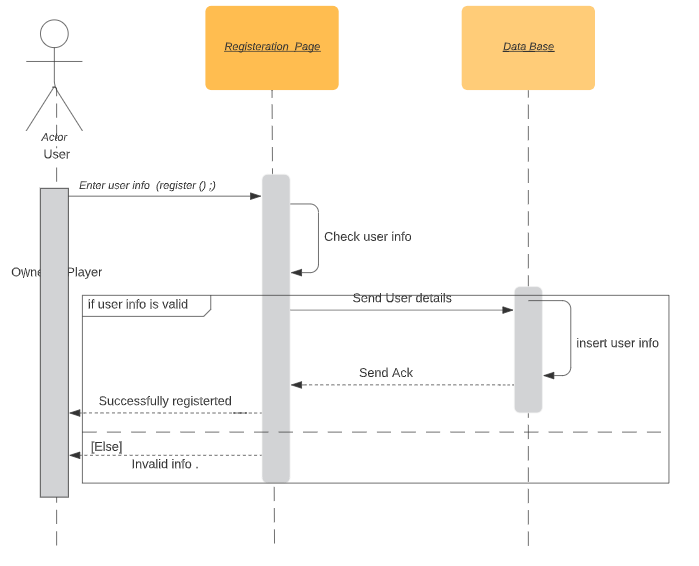
## Sequence diagrams

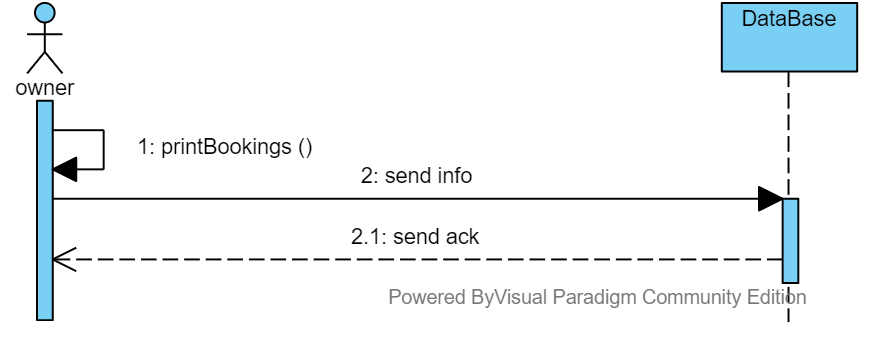












### Class - Sequence Usage Table

| **Sequence Diagram** | **Classes Used** | **All Methods Used** |
| --- | --- | --- |
| 1. Add Playground | Admin  User | login() |
| 1. Approve Playground | PlaygroundOwner | approvePlayground() |
| 1. Book a Playground | Booking | book() |
| 1. Create team | Team | createTeam() |
| 1. Register user | User | tegiser() |
| 1. View playing hours | PlaygroundOwner | printBookings() |

## User Interface Design

|  |  |  |
| --- | --- | --- |
| **Screen ID** | **Screen Name** | **Screen / Wireframe Description** |
|  |  |  |

# Tools

* **Visual-Paradigm**
* **Mocqus**
* **Lucidchart**
* **PlantUML**

# Ownership Report

|  |  |
| --- | --- |
| **Owners** | **Item** |
| Mahmoud Ashraf |  |
| Nader Fikry | Three sequence diagrams, SDS document,  Implementation of Abstract Objects package. |
| Fady Emad |  |