The Chiefs

CS 460W

Project deliverable 1

Part 1

**1.0 - Introduction**

**1.1 - Purpose**

The purpose of this document is to present a detailed description of a system that manages court reservations and financial accounts. The document will explain the functionality and operations of the system, its features and user interface. The document was written for the developers and users of the system.

**1.2 - Scope**

The software system will manage court reservations and financial accounts, as well as keeping track of member information for a tennis club with twelve tennis courts. The system will have a graphical interface and provide login access to membership chair, billing staff (including treasurer), and other members. All data will be managed by a database containing member and membership information, court information, account information, and scheduling.

**1.3 - Definitions, Acronyms, abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| Tennis court | Playable field owned by the tennis club |
| Court Schedule | Schedule of the tennis court used by the members and the club |
| Member | Person that pays membership to the club |
| Guest member | Person invited by a member to the club |
| Financial account | Billing information associated with the member |
| Membership Chair | Staff maintaining membership information |
| Billing Staff | Staff responsible for billing maintenance |
| Database | Collection of all data maintained by the system |
| Software Requirements Specifications | Document describing functionality of the system (such as this document) |

**1.4 - References**

IEEE *IEEE Std 830 – 1998 IEEE Recommended Practice for Software Requirements Specifications*, IEEE Computer Society, 1998

**1.5 - Overview**

Chapter two gives the general overview of the product. It outlines the requirements and provides context for specific requirements outlined in further chapters. Chapter three provides detailed technical requirements, functional requirements, external interface requirements, and performance requirements, establishing functionality of the product. The sections describe the same product but have different audiences.

**2 - General Description**

**2.1 - Product perspective**

The court reservation system is an independent product which handles all members, treasurer, and admin services such as memberships. It will have four types of actors acting with the system: Member, Guest member, Membership Chair (admin), and Billing Staff (treasurer), that access a court reservation system via web interface. Users require limited hardware, just a basic setup with internet connection.

**2.2 - Product functions**

This section outlines the use cases for each actor type interacting with the system.

**2.2.1 – Membership chair creating new account**

Brief Description: The Membership Chair creates a new account for the new member.

Step – by – Step description:

Before this step can be initiated, the Membership Chair needs to log in to the tennis court reservation system.

1. The Membership Chair chooses the create new account option
2. The system displays the option to enter login name, Member name, date of birth, email, address, contact information and initial password
3. The Membership Chair types the information for the new member
4. The Membership Chair presses Create New Account button
5. The system enters the new member information into the database
6. The system displays the message whether the account is created successfully

Summary of the use case:

The new perspective Member supplies their personal and contact information to the Membership Chair on the application. The membership chair takes the information and puts it into the court reservation system. The initial login password for the new member is chosen by the Membership Chair. The Membership Chair will provide the login name and the password to the new perspective member.

**2.2.2** – The Membership chair terminates the membership of an existing member.

Brief Description: The Membership Chair terminates an account for a Member in the system.

Step – by – Step description:

Before this step can be initiated, the Membership Chair needs to log in to the tennis court reservation system.

1. The Membership Chair chooses terminate existing account option
2. The system shows an option to search for a Member username.
3. The Membership Chair types in the name they want to terminate and selects the search button
4. The system displays the list of usernames and asks for confirmation of account deletion
5. The Membership Chair chooses the name and presses the delete button
6. The system removes the member information from the existing user's database
7. The system marks the account terminated and will display a confirmation message

Summary of the use case:

The Membership Chair will terminate an existing account either per request of the Member or for nonpayment of the dues or for other violations of club policy. The system will mark the account as terminated and will notify the member about the termination using existing contact information. The system will remove the member from the “existing user's” database but keep it in the database representing all members.

**2.2.3 – Member updates personal information**

Brief Description: The Member updates their personal information

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on a button Edit Personal Information
2. The system pulls up the current information on the screen
3. The Member changes the information on the screen
4. The Member pushes the Save Information button
5. The system applies changes to the database
6. The system confirms that the changes are saved

Summary of the use case:

The Member will access and modify their personal information without accessing information about any other members.

**2.2.4 – Member updates billing information**

Brief Description: The Member updates their billing information

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on button Update Billing
2. The system pulls up current billing information including credit card and bank account
3. The Member edits the information
4. The Member pushes the Save Information button
5. The system applies the changes to the database
6. The system confirms that the changes are saved

Summary of the use case: The Member accesses their billing information and updates it.

**2.2.5 – Member pays bills**

Brief Description: The Member pays dues and fees

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on button Pay Fees
2. The System will show the current itemized bill
3. The Member selects the form of payment the Member will be using from the billing information list and enters the amount they would like to pay, then clicks Pay the Bill button
4. The system will deduct the amount paid to the total bill and apply change to database
5. The System will display the confirmation that the bill was paid

Summary of the use case: The Member access their bill and pays it.

**2.2.6 – Member creates guest pass**

Brief Description: The Member creates pass for guest

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on button Create Guest Pass
2. The System asks for guest information
3. Member types guest information, clicks on Create Pass button
4. System displays confirmation that pass is created, and emails pass information using member contact information

Summary of the use case: The Member creates guest pass and sends it to email

**2.2.7 – Member books reservation**

Brief Description: The Member books reservation

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on button Court Reservation
2. The System asks for dates, times, and number of people
3. The Member enters desired dates, times, and number of people playing
4. The System shows whether courts are available for desired options
5. The Member confirms reservation
6. The System acknowledges confirmation, emails reservation information to member/ members in reservation

Summary of the use case: The Member books reservation for day and time of their choice

**2.2.8 – Member changes existing reservation**

Brief Description: The Member changes existing reservation

Step – by – Step description:

Before this step can be initiated, the Member needs to log in to the tennis court reservation system.

1. The Member clicks on button Change Reservation
2. The System displays list of current reservations
3. The Member selects reservation, selects Change button
4. The System displays details of chosen reservation
5. The Member either clicks on Cancel Reservation button, Pick a New Date and Time, or Add/Change members and click on Edit Reservation button
6. The System confirms change and sends details of change by email to Member contact information

Summary of the use case:

The Member access current reservation and changes or cancels one of them

**2.2.9 – Membership Chair edits reservation**

Brief Description: The Membership Chair changes existing reservation

Step – by – Step description:

Before this step can be initiated, the Membership Chair needs to log in to the tennis court reservation system.

1. The Membership Chair clicks on button Change Reservation
2. The System displays list of current reservations for all Members
3. The Membership Chair selects Member and reservation, selects Change button
4. The System displays details of chosen reservation
5. The Membership Chair either clicks on Cancel Reservation button or Pick a New Date and Time and click on Edit Reservation button
6. The System confirms change and sends details of change by email to Member contact information

Summary of the use case:

The Membership Chair access current reservation and changes or cancels one of them

**2.2.10 – The Billing Staff runs a billing report**

Brief Description: The Billing Staff runs monthly billing report

Step – by – Step description:

Before this step can be initiated, the Billing Staff needs to log in to the tennis court reservation system.

1. The Billing Staff clicks on button Generate Monthly Billing Report
2. The System displays list of all members and their account balance
3. The System sends list of late and nonpaying accounts to Membership Chair
4. The System sends payment reminder notices to the Members with nonzero balance
5. The System sends monthly statements to all current members

Summary of the use case:

The Billing Staff runs the monthly report. Appropriate information is sent to Members and Membership Chair.

**2.2.11 - View member list**

Brief Description: The Membership Chair or member views list of members in system

Step – by – Step description:

Depending on login credentials, this member list will display active members or a list of all members from the past three years.

1. User clicks on button View Member List
2. The system displays a list of all members excluding account balance and user id. If the Membership Chair is signed in, the list will include user id.
3. System provides search bar for the purpose of looking up specific members

**2.3 - User Characteristics**

The Membership Chair is expected to have access to computer, access to email accounts, and be internet literate.

The Member is expected to be internet literate, have access to a mobile device, and have access to the internet. They are also expected to have an email account and a form of payment, such as a credit card or bank account.

The Billing Staff is expected to be internet literate and have access to a computer.

**2.4 - General Constraints**

Personal and financial information must be protected, stored, and used according to Government regulations. The system will require a computer with storage to handle historical data for club membership and transactions. Remote access to application will require internet access. Members are required to receive emails from club on their mobile devices in order to gain access to the club.

**2.5 – Assumptions and Dependencies**

Appropriate licensing can be acquired for database product capable of handling club transactions. Appropriate middleware technology such as PHP can be installed on the club server. Database communication libraries will be available for middleware product. The front end technology such as a browser is capable of communicating with the middleware. The front end, middleware, and back end technology support necessary security constraints.

**3 – Specific Requirements**

**3.1 Functional Requirements**

This section describes how inputs are processed into outputs within the system. Each function is defined with its purpose, input parameters, processing logic, and expected output. These requirements ensure that the system behaves as expected and allows objective verification of implementation.

**3.1.1.1 Introduction**

The system must allow new members to register by providing their details. Each member will have a unique Member ID generated upon registration. Members should be able to update their contact information and opt in/out of sharing their details in the directory.

**3.1.1.2 Inputs**

Full Name (String, required)

Email (String, required, unique)

Phone Number (String, optional)

Membership Fee Payment Status (Boolean, default: unpaid)

**3.1.1.3 Processing**

Validate inputs (check for missing or duplicate email).

Generate a unique Member ID.

Store the member details in the MySQL database.

Allow updates to phone number and email.

**3.1.1.4 Outputs**

Confirmation message upon successful registration.

Error message for duplicate email or invalid input.

Updated details if member modifies their information.

**3.2.1 User Interfaces**

The Club Management System will provide a graphical user interface (GUI) for members, administrators, and the treasurer. The interface will be intuitive, easy to navigate, and support the following functions:

Key Characteristics of the User Interface

Screen Layout:

Dashboard: Displays upcoming bookings, guest limits, and billing status.

Booking Screen: Allows users to select a court, time, and players.

Member Directory: Lists active members (with opt-in privacy settings).

Billing Section: Displays annual dues, guest charges, and payment status.

Admin View: Shows past members, financial reports, and court usage history.

Function Keys & Buttons:

"Book Court" – Opens booking form.

"Add Guest" – Allows members to register guests.

"Pay Bill" – Redirects to the billing section.

"Update Profile" – Allows members to modify personal details.

Help Functions:

Tooltips for booking rules and membership guidelines.

FAQ Section explaining club policies and billing rules.

Verifiable Learning Requirements:

New members should be able to make a reservation within 5 minutes of using the system.

Admins and the treasurer should be able to access financial data within 3 minutes of logging in.

**3.2.2 Hardware Interfaces**

The system is designed to be accessed via web browsers or a desktop application. It must support:

Screen Requirements:

Minimum resolution of 1024x768 for proper display.

Optimized for laptops, desktops, and tablets.

Input Devices:

Keyboard and Mouse for desktop access.

Touchscreen support for tablet access.

Storage Requirements:

The MYSQL database will be hosted on a local server, requiring at least 1GB of available storage.

Processing Power:

Minimum Intel Core i3 or equivalent processor.

At least 4GB RAM for smooth performance.

3.2.3 Software Interfaces

The Club Management System will depend on the following software components:

Operating Systems Supported:

Windows 10/11

macOS

Linux distributions (Ubuntu, Fedora)

Programming Language:

Java 17 or higher

Database:

MYSQL (Lightweight, no external database server required)

Frameworks & Libraries:

Java Swing (for GUI)

JDBC (Java Database Connectivity) for MYSQL integration

Third-Party APIs (if needed):

JavaMail API for sending billing and booking confirmation emails.

Interoperability with Other Applications:

The system may export billing reports as CSV or PDF for financial review.

Future integration with payment gateways (e.g., PayPal, Stripe) is possible.

**3.2.4 Communication Interfaces**

Although the system is primarily local, certain operations may require internet communication for billing and notifications.

Internal Communication:

The system will use local database queries for fetching member, booking, and payment data.

Admins and treasurers will have role-based access to member history.

Email Notifications:

The system will use SMTP (Simple Mail Transfer Protocol) to send billing invoices and booking confirmations.

Future Network Expansion (Optional):

If extended to a web-based version, it will use HTTP/HTTPS protocols.

**3.3 Performance Requirements**

The system must have the capability to handle multiple user requests with smooth performance. Below are the performance requirements:

1. **Concurrent users:**

System should be capable of supporting 50 concurrent users atleast without performance degradation.

1. **Response Time:**

Under normal load system response time should be within 2 seconds for any user action.

1. **Database Transactions:**

The system should maintain data integrity and should be able to handle around 100 transactions per second.

1. **Scalability:**

The system should be scalable and should be expandable to support more users and courts in the future.

**3.4 Design Constraints**

Design constraints are imposed according to the requirements of the software, hardware requirements and industrial standards.

**3.4.1 – Standard compliance**

1. The system must comply IEEE software engineering standards for the requirements of the software.
2. It should follow the best practices of security and for sensitive data it should include encryption.
3. The system must support the data protection of the user’s information.

**3.4.2 – Hardware limitations**

1. The system should run on standard systems like Windows, Linux or macOS.
2. The system should run on the machines having 4GB RAM at least and dual core processors.
3. For mobile devices the system should smoothly run on the devices with 2GB ram.

**3.5 System attributes**

**3.5.1 – System availability**

1. System should have up time of 99.5% and should ensure minimum possible down time.
2. There should be a restart mechanism, so system can recover from failure automatically.
3. There should be automatic back up of database in case of data-base failure.

**3.5.2 – Security**

1. Multifactor authentication should be used for user authentication
2. All sensitive data should be encrypted
3. Access to the system should be role based to ensure that only authorized user can perform a specific operation.

**3.5.2 – Maintainability**

1. System should be updateable in modular way so core functionality should not be affected.
2. Objected oriented principles should be followed so that debugging should be easy.
3. Errors should be logged by using monitoring tools to track system performance

**3.6 - Other requirements**

1. There should be mobile friendly interface as well for reserving courts from smart phones
2. Email or SMS notifications should be integrated in the system for court booking confirmations.

**Minutes**

The Chiefs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Jacob Tsekhansky  (Team Lead) | Evan Riddick | Talal Alsmait | Cody Beard(Recorder) |
| 1/22(class) | 75min | 75min |  | 75min |
| 1/27 (class) | 75min | 75min | 75min | 75min |
| 1/29 (class) | 75min | 75min | 75min | 75min |
| 2/3 (class) | 75min | 75min | 75min | 75min |
| 2/5 (class) | 75min | 75min | 75min | 75min |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Meeting time – [Meeting Link](https://teams.microsoft.com/l/meetup-join/19%3ameeting_NGQyODM0ZDMtYWZlZS00OGQ1LTkzZjktNTkwOWM2NTExM2Yw%40thread.v2/0?context=%7b%22Tid%22%3a%22b78b91e8-428e-422f-a5b4-a8f6cab07104%22%2c%22Oid%22%3a%227e4de901-523d-47e0-bf0c-eeac27174d39%22%7d) Thursdays 5:00 PM – 6:00 PM (Password: RK9VT2NL)

**Deliverable 1(Action items):**

Cody Beard (Recorder) – Responsible for section 3.0-3.2.4. Also responsible for accounting minutes for each team member

Talal Alsmait - Responsible for section 3.3-3.6

Jacob Tsekhansky (Team Lead) – Created Outline for SRS Doc as well as took a part in sections 1.0 - 2.5

Evan Riddick – Responsible for section 2