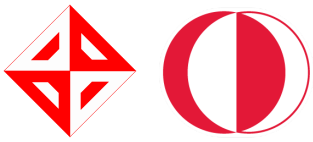

# MIDDLE EAST TECHNICAL UNIVERSITY

# COMPUTER ENGINEERING

## CENG 49X COMPUTER ENGINEERING DESIGN

## CRDT DOCUMENT


TEAM NAME: HOPPERS

Zehra HAYIRCI

Ceren DİKMEN

Tuğçe YILMAZ

Ülkem KASAPOĞLU

**Supervisor** : Assoc. Prof. Uluç SARANLI **Advisor**: Itır ÖNAL

# 1. Introduction

This software requirements document specification provides complete information about the system called Smart Bookmark  which will be developed by our project team Hoppers. The system is planned as a smart bookmark device for print books and a mobile application. In this section, we are going to give the definition of the problem, introduction of the purpose and scope of this document, definitions, acronyms and abbreviations, references and overview. In the following sections, we are going to introduce an overall description and features of the project, present the specific requirements, use cases, data models and behavioral models and their detailed description. Finally, as our development program, we are going to state planning, team structure, team schedule and conclusion of the project respectively.

# 2. System Requirements

## 2.1. Information About Initial and Revised System Requirements

### 2.1.1. Bookmark Device Functional Requirements

#### 2.1.1.1. Functional Requirement 1

Device should transfer sensor data to the Android application via bluetooth.

#### 2.1.1.2. Functional Requirement 2

Device should listen the messages coming from to the Android application via bluetooth.

### 2.1.2.  Web Server Functional Requirements

#### 2.1.2.1. Functional Requirement 4

Web server should handle multiple stream request and user inputs efficiently.

#### 2.1.2.2. Functional Requirement 5

Web server should keep user data(username,password,personal information such as birthdate,bio,location etc.), user's friend list and books of user data on database.

#### 2.1.2.3. Functional Requirement 6

Web server should authenticate user connection request.

#### 2.1.2.4. Functional Requirement 7

Web server should send user info to Android local database, also book details, books of user when requested.

#### 2.1.2.5. Functional Requirement 8

Web server should add books to related data table when requested by Android application.

#### 2.1.2.6. Functional Requirement 9

Web server should edit user's profile information when requested by user via Android application.

#### 2.1.2.7. Functional Requirement 10

Web server should store statistical values i.e. how much one read, how many minutes it took to read one page on the average.

#### 2.1.2.8. Functional Requirement 11

Web server should support both online and offline features. Therefore, synchronization in both way should be implemented (server to application local db, application local db to server).

### 2.1.3. Android Application Functional Requirements

#### 2.1.3.1. Functional Requirement 12

When there is not any user logged in, application should ask for username and password. User can choose register or log in existing account.

#### 2.1.3.2. Functional Requirement 13

User can edit his/her own profile information via application. Application should send edit request to web server.

#### 2.1.3.3. Functional Requirement 14

Application should show current book information and statistics of the user about that book.

#### 2.1.3.4. Functional Requirement 15

Application should allow camera scanning and when an ISBN value is detected, it should find and get book information from Google Books database.

#### 2.1.3.5. Functional Requirement 16

User can add the book which fetched form Google Books by ISBN scanning, and they should be shown on the ‘Bookshelf’ tab.

#### 2.1.3.6. Functional Requirement 17

Application should send a notification when current book is not opened in a specified time by user.

#### 2.1.3.7. Functional Requirement 18

User can give information about the reading process i.e. pause book, quit book.

#### 2.1.3.8. Functional Requirement 19

User can add quotes from the book by manually or by taking photo of it. When user wants to see those quotes, they should be shown on the ‘Quotes’ tab under the ‘Current Book’ section. User can share quotes on Social Media whenever s/he wants.

#### 2.1.3.9. Functional Requirement 20

User can start reading a new book after finishing the current book.

#### 2.1.3.10. Functional Requirement 21

Application should show friend list and one’s friend information such as user information of that friend, current book of him/her.

#### 2.1.3.11. Functional Requirement 22

After a book is finished by the user, application should ask for a rate to user. By analyzing rating values, application should recommend more books to the user.

#### 2.1.3.12. Functional Requirement 23

User can pause a book, and continue after a while.

### 2.1.4. Non-functional Requirements

#### 2.1.4.1. Usability

Since user interfaces of the project does not have complex tasks on them, user should have confidence using those easily.

Usability requirements of the system are:

- Necessary configurations of fixed device should be automatic not to confuse user.
- Transition between tabs on application should be arranged properly.
- Statistical information should be simplistic and clear.
- The user interface elements (e.g. login,pause book, add book) should be easy to understand.
- Bookmark device should be compact and easy to transport.

#### 2.1.4.2. Reliability

The reliability of the system depends on three main tools: Bookmark device,web server and Android device.

- Since bookmark device tracks reading statistics periodically, in case of lack of wireless connection.
- If any user has more than one device, application should list each one of them and their related books separately, and server should store them in its data tables.
- The reliability of the web server has also a crucial importance. Web server should be 100% percent available.
- Android device should send web server to data both periodically and user-controlled.
- It is expected that sensors, bluetooth module and other tools works perfectly until their lifetime expires.

#### 2.1.4.3. Performance

- System should work real-time which means there should be an acceptable time delay.
- Web server should be able to handle multiple device and user connection.
- Image capturing, transferred data size, speed of connection, response time, processing speed must be considered.

#### 2.1.4.4. Supportability

- The system needs to be cost-effective to maintain. Therefore, bookmark device should have no non-essential tools like camera.
- Class libraries that will be used contain but is not limited to are libraries for ISBN parsing, OCR, designing user interface and machine learning. Libraries can be extended for future needs.

#### 2.1.4.5. Security

- Designing the client-server application requires the security measurements. To maintain a reliable system, server should remain opened. However, it makes the system vulnerable against internet attacks. A tool called ***Fail2ban*** is installed to server. It scans

log files and bans IPs that show the malicious signs -- too many password failures, seeking for exploits [2]. Then data can be stored in server securely.

## 2.2. Information About Initial and Revised Project Plan

The purpose of SmartBookmark project is to make reading printed books more attractive for people of all ages and to solve the problem of readers arising from time constraints. This project involves an embedded bookmark device consisting of a microcontroller and sensors, in addition to a mobile application. The device is a clip mechanism and used by being attached around a printed book. It is connected to a mobile phone via Bluetooth, and sends the reading data to the Android application where a reader can follow his/her reading progress visually. Therefore, the readers can track how much and how fast they read. They can also get notifications when the reading goal is not reached. Furthermore, many features are offered in the mobile application, such as personalized book recommendations, bookshelf in mobile, saving and sharing quotes from books a reader likes, social platform enabling to follow bookshelves of friends. All of which help a reader to commit a book more strongly. The last but not least, the SmartBookmark is a way of improving reading experience and increasing reading quality.

### 2.2.1. Initial Project Plan

**Some suggestions that initially suggested as solutions were:**

- Bookshelf for readers
- Measuring reading progress and reading speed, keeping them as statistics
- Notification system for reminding users to read
- Book recommendation and sharing platform
- Quote extraction from books and sharing among social media

We thought about using light sensor for measuring time and sending notification. However at the beginning we are unsure about sensors for measuring reading progress.

### 2.2.2. Revised Project Plan

All of the features above are implemented in the system. Also, we have extra modifications.

**Solutions in the revised project plan:**

- Adding book to system has been revised and isbn barcode reading implemented. Readers also add isbn manually.
- Social media has been implemented for following bookshelves and reading progress of friends. We have initially thought social media as a book sharing platform, however, social media feature has a mission for influencing people by showing books of friends.
- Potentiometer is used for a solution for measuring progress in the book. Reading pace and approximate book finishing time are calculated with the help of new sensor.
- Support for reading multiple books at the same time is implemented to the system. Readers can pause their current book and return to them whenever they want without losing time and progress data.

- Support for both online/offline features in mobile application is provided. When user offline, for example, can start/finish/pause/continue a book, add quote, get recommendations.

## 2.3. Use Case Diagram

| **Use Case** | **Description** |
|---|---|
| Register | Users need to register themselves and their devices to system. |
| Login | Users need to login to start using system. |
| Add Device | Users can add a bookmark device to the account. |
| Remove Device | User can remove the bookmark device from account. |
| View Profile | Users can see their profile. |
| Edit Profile | Users can change their personal information. |
| View Current Book | Users can see their current read book and its reading statistics. |
| Synchronize application | Users can refresh the screen to see new incoming data. |
| Add Quotes | Users can add quotes which they like. |
| View Saved Quotes | Users can view saved quotes. |

| | |
|---|---|
| Share Saved Quotes | Users can share quotes with their friends via their account. |
| Add Book | Users can add a new book to the wished books list. |
| View Books | User can view added books. |
| Start Reading Book | User can start a book by informing the application about this. |
| Pause Reading Book | User can pause a book and send it to wished book list back, by entering dropped page. |
| Stop Reading Book | User can finish reading a book. |
| Rate Book | User can rate a book after finishing it. |
| Get Reminder Notification | User can get a reminder notification when s/he doesn't read within fixed time. |
| Add Friends | User can add friends. |
| View Friends' Dashboards | User can view her/his friends' dashboards. |
| View Recommendations | User can get recommended books. |
| Edit Reminder Notification Settings | User can switch on/off reminder notifications and edit the time period of reminder notifications. |
| Logout | User can logout system. |

**Primary actors: User  Secondary actors: None**

**Active actors = User  Passive actors = None**

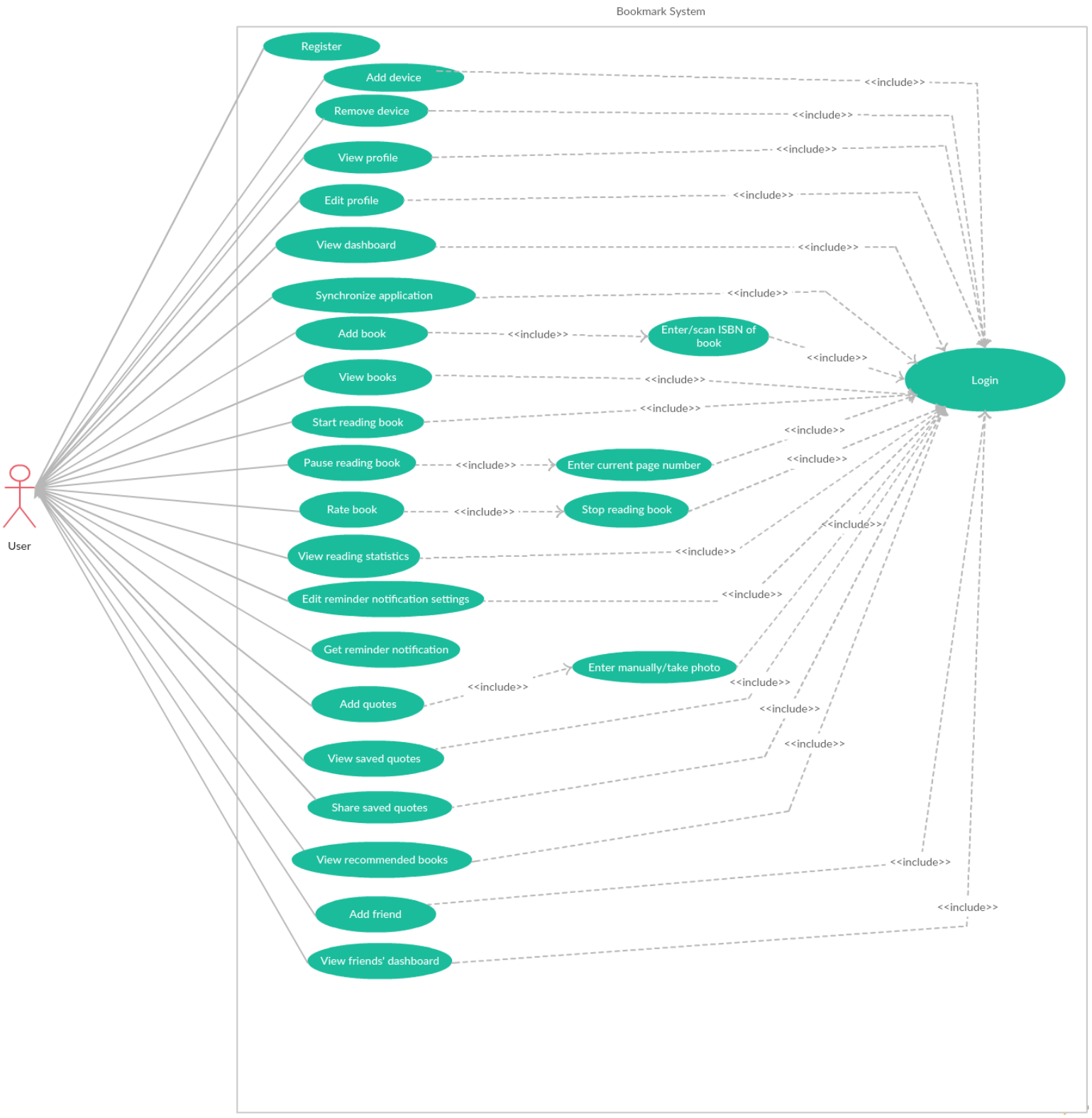
Bookmark System
Register
Add device
Remove device
View profile
Edit profile
View dashboard
Synchronize application
Add book
Enter/scan ISBN of book
View books
Start reading book
Pause reading book
Enter current page number
Rate book
Stop reading book
View reading statistics
Edit reminder notification settings
Get reminder notification
Enter manually/take photo
Add quotes
View saved quotes
Share saved quotes
View recommended books
Add friend
View friends' dashboard
Login
User
<<include>>

**Figure 1**

### 2.3.1. Register Use Case

| | |
|---|---|
| Description | This use case describes the operation that new users create a new account |
| Actors | New users |
| Precondition | - |
| Trigger | |
| Basic Flow | New users enter their<br>new username<br>Email<br>password<br>Gender<br>location<br>And click on “Register” button on register menu |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.2. Login Use Case

| | |
|---|---|
| Description | This use case describes the operation that new users login the system |
| Actors | User |
| Precondition | User already has an account |
| Trigger | User clicks on “I have an account” button |
| Basic Flow | 1- user enters their<br>username/email<br>password<br>2- user clicks on “Login” button |

| | |
|---|---|
| Exception Flow | - |
| Post Conditions | Email/password check. If they are incorrect they are redirected to login page. If they are correct, the user is directed to main page of the application |

### 2.3.3. Add Device Use Case

| | |
|---|---|
| Description | This use case describes the operation that user pairs their bookmark to their mobile application |
| Actors | User |
| Precondition | User should have paired the bookmark and mobile device, before pairing the bookmark the mobile application. |
| Trigger | User clicks “Add Device” button in the settings |
| Basic Flow | 1- User clicks the button<br>2- Bluetooth searches for the devices<br>3- User clicks the Device that is listed<br>4- Mac address of the selected device is saved in database(or stored in variable)<br>5 - “Add device” button is disabled, device becomes visible.<br>6- Whenever user login and request for bluetooth connection, saved mac address is connected automatically |
| Exception Flow | - |
| Post Conditions | Mac address of the bookmark is saved in local |

### 2.3.4. Remove Device Use Case

| | |
|---|---|
| Description | This use case describes the operation that user unpairs their bookmark to their mobile application |
| Actors | User |
| Precondition | User should have a added device already. |
| Trigger | User clicks “Remove Device” button in the added device page in directed from settings? |
| Basic Flow | 1- User clicks the button<br>2 - A message window pops up which asks “Are you sure to remove this device?”. If user allows, remove action is done and user is directed to profile page.<br>3- “Add device” button is enabled, device become invisible.<br>4- Mac address of the removed device is deleted from database (or nulled in case storing a variable). |
| Exception Flow | - |
| Post Conditions | Mac address of the bookmark is saved in local |

### 2.3.5. View Profile Use Case

| | |
|---|---|
| Description | This use case describes the operation that users views their profile |
| Actors | User |
| Precondition | User has to login the system |
| Trigger | User clicks on “Profile” icon |

| | |
|---|---|
| Basic Flow | 1-User clicks on the icon<br>2-User is directed to Profile page |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.6. Edit Profile Use Case

| | |
|---|---|
| Description | This use case describes the operation that users edit their profile page |
| Actors | User |
| Precondition | User has to login the system and click on the Profile icon |
| Trigger | User clicks on any info in the profile page |
| Basic Flow | 1- User clicks on information<br>2- User enters new information<br>3- User approves the change |
| Exception Flow | User enters an incorrect info<br>User info does not match the requirements |
| Post Conditions | New data is saved in database |

### 2.3.7. View Dashboard Use Case

| | |
|---|---|
| Description | This use case describes the operation that users view dashboard |
| Actors | User |
| Precondition | User has to login the system |

| | |
|---|---|
| Trigger | User clicks on “Dashboard” icon on the bottom navigation menu |
| Basic Flow | 1- User clicks on icon<br>2- User is directed to Dashboard fragment.<br>3- User name ,profile image, info attributes are shown at top of the page. Also overall statistics are shown, which are number of the read books, number of read pages and total time spent on the books.<br>4- There will be 4 tabs below profile section.<br>- “Current Books” where s/he can view current book information, reading progress on current book.<br>- “Finish Books” where s/he can view list of read books up to now , and their statistics.<br>- “bookshelf” where s/he can view list of bookshelf added before but not read yet. User can also request new recommendations via a button.<br>- “Quotes” where s/he can see list of all added quotes. |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.8. Synchronize Application Use Case

| | |
|---|---|
| Description | This use case describes the operation that users can refresh the screen to see new incoming data. |
| Actors | User |
| Precondition | User has to login the system and in the home tab. |
| Trigger | User refreshes dashboard |
| Basic Flow | 1- User refreshes the dashboard<br>2- If there is an update from device or server, dashboard and local database updates themselves<br>3- If user doesn’t synchronize the application for a long time, a warning message arrives to user, which informs about that synchronization is needed. |

| | |
|---|---|
| Exception Flow | Bluetooth and web connection is not allowed on mobile device |
| Post Conditions | If there are any, updates are shown |

### 2.3.9. Add Quotes Use Case

| | |
|---|---|
| Description | This use case describes the operation that users add quotes from the book |
| Actors | User |
| Precondition | User has to login the system |
| Trigger | User clicks on add icon on the bottom navigation menu |
| Basic Flow | 1- User clicks on icon<br>2- User is directed to add fragment<br>3- User clicks on “Add Quote”<br>4- A window pops up which offers two options “From current book” and “From other book”.<br>5 - After user decides on the option, a new window pops up, which asks for “book name” and “page number”. If user have selected “current book”, book name is automatically fetched from database. However, if user have selected “other book”, user has to enter the book name manually. In both cases, user also should enter the page number.<br>4- User scans the book quote<br>5- This data is sent to server<br>6- Server returns the extracted text<br>7- This text, date, book name and book page is saved to be shown in Quotes tab in dashboard.<br>8 - Quote is saved in database, Quotes table. |
| Exception Flow | - web connection is not allowed on mobile device<br>- Camera does not work properly |
| Post Conditions | - |

### 2.3.10. View Quotes Use Case

| | |
|---|---|
| Description | This use case describes the operation that users view their extracted quotes |
| Actors | User |
| Precondition | User has to login the system and in the library tab |
| Trigger | User clicks on “Quotes” tab in dashboard. |
| Basic Flow | 1- User clicks on “quotes”<br>2- User views extracted quotes ordered by date<br>3- User can filter quotes by book name. |
| Exception Flow | |
| Post Conditions | - |

### 2.3.11. Share Quotes Use Case

| | |
|---|---|
| Description | This use case describes the operation that users share quotes from the book |
| Actors | User |
| Precondition | User has to login the system and in the home tab |
| Trigger | User clicks on quotes |
| Basic Flow | 1- User clicks on quotes<br>2- User clicks on the “share” icon near the book that is going to be shared<br>3- User chooses the social media or application |
| Exception Flow | - web connection is not allowed on mobile device |
| Post Conditions | - Quote is sent to other application |

### 2.3.12. Add Book Use Case

| | |
|---|---|
| Description | This use case describes the operation that users add new book |
| Actors | User |
| Precondition | User has to login the system |
| Trigger | User clicks on “Add ” icon on the bottom navigation menu |
| Basic Flow | 1- User clicks on add book<br>2- User is asked to scan the isbn barcode or enter manually isbn author and name of the book<br>3- If there is internet connection, isbn is searched and data is saved in the database<br>4- If there is no internet, the book object is created but it has to be checked when the connection provided |
| Exception Flow | |
| Post Conditions | - Book is added to bookshelf |

### 2.3.13. View Books Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can view the added books in “Dashboard? Profile? Library” tab of application. |
| Actors | User |
| Precondition | User has to login the system |
| Trigger | User clicks on the "Dashboard” tab on the bottom navigator. |
| Basic Flow | 1 - User can click on Current Books, bookshelf or Read Books from 4 tabs appears on the screen.<br><br>2 - User can view the basic information (e.g name, author, publish year, cover image) of the books under each of the categories. There are also buttons for specific tasks near the book records. |

| | |
|---|---|
| | 3 - Books are currently read are listed in “Current Books”. User can have more than one current book on the condition that pausing others while reading a specific one. User can see progress of current book, which contains reading or non reading time, how much of the book is left. Pause button is also located near each current book record.<br><br>4 - Books which are already added but not read yet are listed below bookshelf. Books which are added and read, are listed below Read Books.<br><br>5 - Near each “Wish book”, “Start now” button appear. There is a ”recommend me a new book” bottom of the page to get new recommendations.<br><br>6 - Near each “Read book”, “View statistics” button appear. User can see “start time”,” finish time” of the book, “spend time” in the book, and some graphs which shows the progress to the user. |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.14. Start Reading Book Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can start reading a book. |
| Actors | User |
| Precondition | Related book should have been added to bookshelf list. |
| Trigger | User clicks the “Start now” button near the book record in WishBookLibrary tab. |
| Basic Flow | 1 - A message is sent to smart bookmark to activate it.<br><br>2 - When smart bookmark gets the message, starts to send data to the application about reading information.<br><br>3 - After starting a book, it becomes the “current book”. User can |

| | |
|---|---|
| | see it in “Current book” tab in dashboard |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.15. Pause Reading Book Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can pause reading a book. |
| Actors | User |
| Precondition | Related book should be the current book. |
| Trigger | In “Dashboard” Tab, user clicks on “Pause book” button near the current book record. |
| Basic Flow | 1 - A window pops up and last page is requested from the user.<br><br>2 - User enters the last page. It is saved in database.<br><br>3 - Paused book continues to be shown in “current books” tab, however if a specified time passes, it becomes a wished book again. The record is sent to wished books. Below the “wish book” tab, paused books are listed firstly. bookshelf are shown after the paused books.<br><br>4 - A message is sent to smart bookmark. Therefore, bookmark paused and doesn’t send data until the book becomes current book again. Last reading data is saved to database. And bookmark data is reset. |
| Exception Flow | Last page is not entered. |
| Post Conditions | - |

### 2.3.16. Stop Reading Book Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can stop reading a book. |
| Actors | User |
| Precondition | Related book should be the current book. |
| Trigger | In “Dashboard” Tab, user clicks on “Stop book” button near the current book record. |
| Basic Flow | 1 - A window pops up and “Are you sure to stop this book?” is asked to ensure the request of the user.<br><br>2- If user submits, the book won’t be the current book no longer. Book is seen as a “read book” Library tab. The record is sent to books.<br><br>3 - A stop message is sent to smart bookmark. Then, smart bookmark won’t send reading data no longer.<br><br>4 - User is asked to rate book if now or later. If user rates the book, it is shown in finished books list, near the related book. If user doesn’t rate at that time, rate now button appears near the book record. Rating is also saved to database.<br><br>5 - All reading data is processed and saved as statistics of related book. |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.17. Rate Book Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can rate a book. |
| Actors | User |

| | |
|---|---|
| Precondition | User should have stopped the book. |
| Trigger | 1 - User clicks on “Finish book” button, just after this action, a rating window pops up.<br><br>2 - If user doesn’t rate the book just after finishing it, user can rate via finished books list. First, user finds the related book from the list and clicks on “Rate now” button. Then user completes the rating action. |
| Basic Flow | Rating is done via 5 stars. When user completes rating action, it is saved to database. |
| Exception Flow | - |
| Post Conditions | - |

### 2.3.18. Get Reminder Notification Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can get a notification in case he/she doesn’t read the book within a fixed time. |
| Actors | User |
| Precondition | User should have set the notification settings and started to read a book. |
| Trigger | If user doesn’t read the current book within fixed time, time notification comes in the application. User should have paired bookmark device and application before. |
| Basic Flow | 1 - When user has a break to read the current book, we assume that he/she puts the bookmark over the dropped page, within the book.<br><br>2 - The light sensor detects that light amount decreases and system increments non-reading time until light amount rises again. Device sends the non-reading time data to the application.<br><br>3 - If the time specified by the user is exceeded, a notification appears in the application. |

| | |
|---|---|
| Exception Flow | - |
| Post Conditions | - |

### 2.3.19. Edit Reminder Notification Settings Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can edit read notification settings. |
| Actors | User |
| Precondition | User has to be in Settings |
| Trigger | User clicks on “Edit Reminder Notification” on settings menu. |
| Basic Flow | 1 - User can switch on/switch off the notifications.<br><br>2 - After switching on, user chooses time period of notifications. |
| Exception Flow | |
| Post Conditions | Bookmark measures this fixed time and sends time data to the application. If user doesn’t read during the time specified by user, a notification is sent to the application. |

### 2.3.20. Add Friends Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can have the friends via the application. |
| Actors | User |
| Precondition | User has to login and go to Friends tab on the application |
| Trigger | User clicks on “Add Friend” button |
| Basic Flow | 1- User enters user name of requested person.<br>2- If the person is found, user can add this friend. |

| | |
|---|---|
| Exception Flow | - |
| Post Conditions | Friendship pair is added to database. |

### 2.3.21. View Friends' Dashboards Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can view his/her friends' dashboards |
| Actors | User |
| Precondition | User has allready at least one friend. User in the Friends tab |
| Trigger | User clicks on his/her friends that s/he wants to view |
| Basic Flow | 1- User clicks on his/her friend<br>2-User is directed to friend's dashboard. |
| Exception Flow | |
| Post Conditions | |

### 2.3.22. View Recommendations Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can get new book recommendations. |
| Actors | User |
| Precondition | User should have read at least 20 books and their ratings before. |
| Trigger | User clicks on "Recommend me a new book" button in the "Wish book" tab. |
| Basic Flow | 1- Users ratings are evaluated. Genre and author of user's favourite books are classified with machine learning algorithms.<br>2- According to this results, new books are |

| | |
|---|---|
| | recommended to the user. |
| Exception Flow | No enough rating |
| Post Conditions | |

### 2.3.23 Logout Use Case

| | |
|---|---|
| Description | This use case describes the operation that user can logout his/her account |
| Actors | User |
| Precondition | User in the Settings |
| Trigger | User clicks on Logout |
| Basic Flow | User clicks on logout and s/he is logout account on application |
| Exception Flow | |
| Post Conditions | User account is logged out |

# 3. System Design

## 3.1. Module Structure

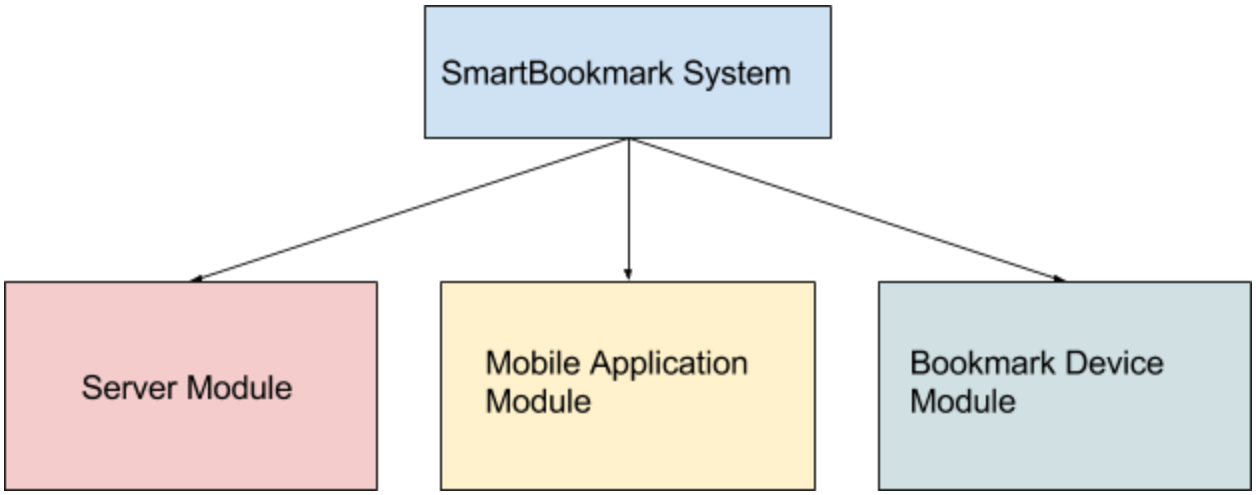


**Figure 2**

## 3.2. Source Code Structure

### 3.2.1. Bookmark Device Source Code Structure

<ArduinoTest>

serialCommunication

serialCommunication-v1

serialCommunication-v2

serialCommunication-v3

serialCommunication-v4

serialCommunication-v5

### 3.2.2. Mobile Application Source Code Structure

Classpath to java classes that are in main app in git:

<SmartBookmarkTest>

app/

src/main/java/

com/example/user/

activity/ClassName

src/main/res/

layout/

activity_ClassName

### 3.2.3. Server Source Code Structure

Classpath to classes that are in git:

<ServerTest>

myapp

admin.py

models.py

serializers.py

...

myproject

settings.py

urls.py

…

## 3.3. Component Diagram

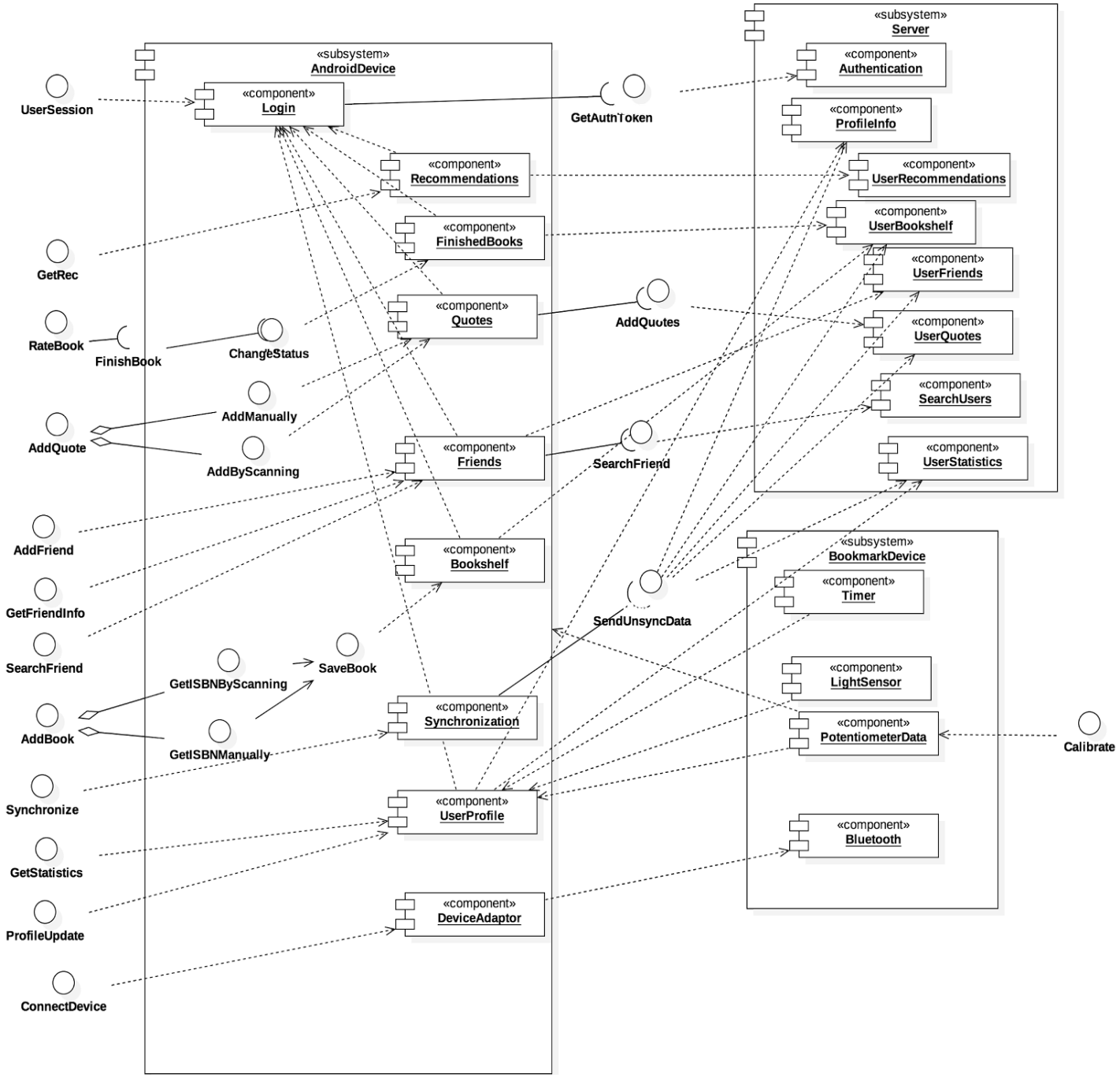


**Figure 3**

## 3.4. Deployment Diagram

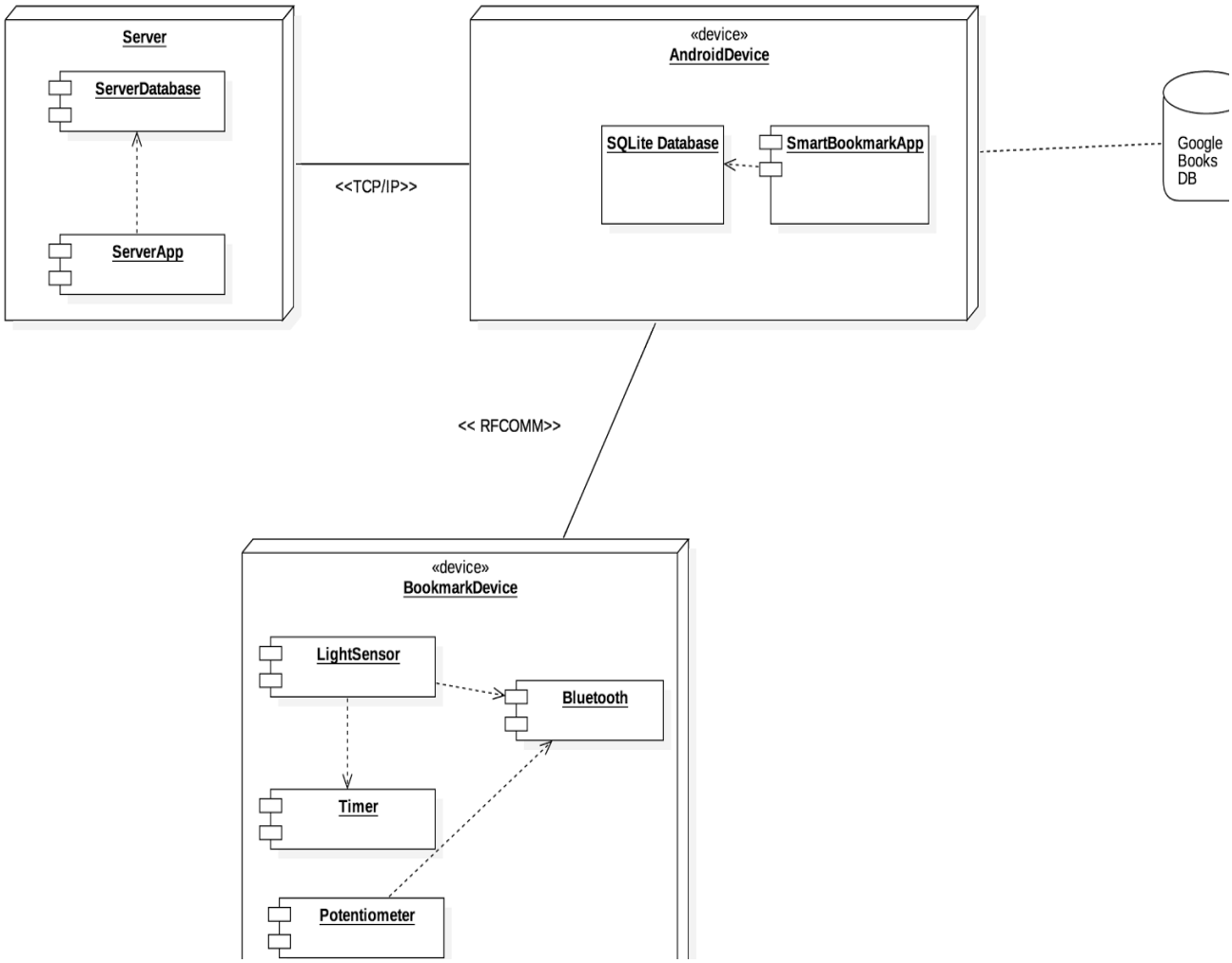


**Figure 4**

# 4. Testing

## 4.1. Test Plan and Scenarios

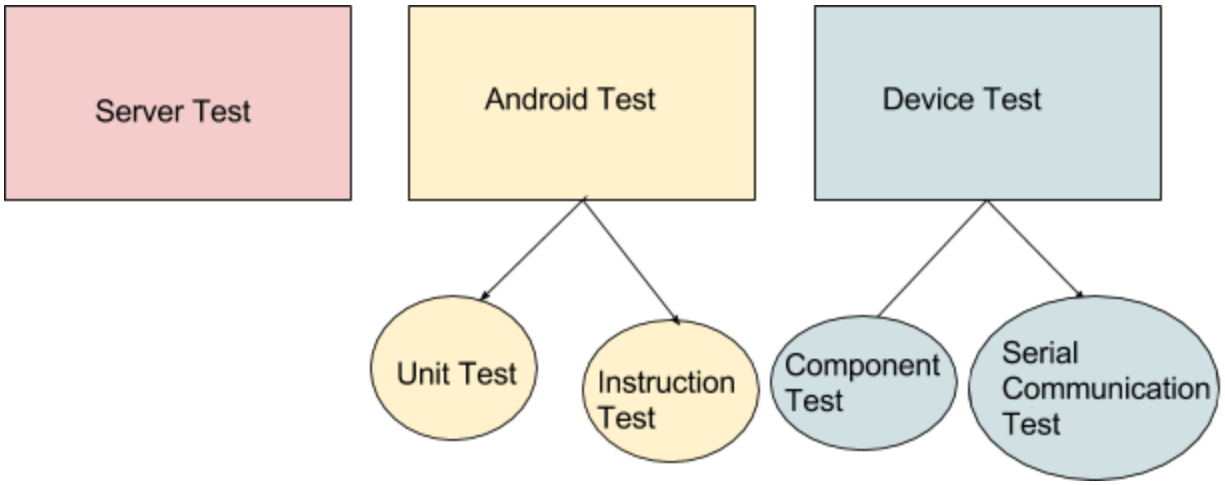


**Figure 5**

### 4.1.1. Bookmark Device Test Plan

#### 4.1.1.1. Hardware Components Test Scenarios

We made hardware development in Arduino Board with ATMega 328p microprocessor, by using Arduino IDE. We integrated our circuit elements which are sensors and bluetooth module to the board. We firstly tested whether bluetooth module and sensors work properly.

The table shows that the components and expected situations in each test scenario.

| **Component** | **Expectation** |
|---|---|
| HC-05 Bluetooth Module | In unpaired case: LED on module is blinking all the time<br>In paired case: LED on the module is blinking once in 2 second |

| | |
|---|---|
| LDR (Light Sensor) | Sensor output values written in serial Port change with light conditions |
| Potentiometer (Angular Displacement Sensor) | Sensor output values change with angular displacement |

#### 4.1.1.2. Serial Communication Test Scenarios

To provide communication between Android application and embedded device, required tasks were specified in Round Robin Scheduling. These tasks are triggered by UI components of the mobile application. A service runs in background and listens the messages from the embedded device. It is a duplex communication, therefore messages come from the mobile application are also listened in Round Robin loop. As a result, serial communication is done via Bluetooth protocol.

| **Scenario** | **Round Robin Tasks** | **Trigger In Android** |
|---|---|---|
| Start Book | ListenTask | startButton |
| Calculate Reading/Non-reading Time | LightTask | -<br>(Triggered in Round Robin itself) |
| Pause Book | ListenTask | pauseButton |
| Continue Book | ListenTask | continueButton |
| Finish Book | ListenTask | finishButton |
| Calibrate | ListenTask<br>PotTask | measureReadingRatioButton |

| | | |
|---|---|---|
| Send Data | SendTask | refreshButton |

## 4.1.2. Android Test Scenarios

### 4.1.2.1. Unit Test Scenarios

Unit tests provide us to validate our classes and functions used within them. We tested our *'value-returning'* functions by these tests:

| **Test Class** | **Test Function** | **Expected Result** |
|---|---|---|
| AccountHelperTest | getTokenKey_isCorrect | "Correct" |
| AccountHelperTest | getServerAddress_isCorrect | "Correct" |
| BookAdaptorTest | updateStatusTest | "Correct" |
| BookAdaptorTest | updateDropTest | "Correct" |
| BookAdaptorTest | updateIsSyncTest | "Correct" |
| OCRAdaptorTest | updateIsSyncTest | "Correct" |
| ReadingRatioAdapterTest | getStartMeasureTest | "Correct" |
| ReadingTimeAdaptorTest | sumTimeDataTest | "Correct" |

### 4.1.2.2. Instrumental Test Scenarios

An instrumentation-based test class allows us to send key events to the application interface under test. We have tested our UI components and key activities by these tests.
We have used Espresso for testing our UI components. We have defined it in project dependency. Then, for each activity that have UI components, we have created a test class.

### 4.1.3. Server Test Scenarios

Mainly, end-to-end tests were done to check whether end points work properly or not, by throwing requests in both correct and false cases. Then, it was checked that if expected results came or not. Django test modules were used during tests.

We also checked if expected modifications occured in database or not. During testing, a test database is created in each run and it is destroyed at the end of test. Run command:

$ python manage.py test tests/

## 4.2. Testing Code Structure

### 4.2.1. Bookmark Device Code Structure

Classpath to Arduino test class in git:

<ArdunioTest>
    serialCommunication
        serialCommunicationTest

### 4.2.2. Mobile Application Code Structure

#### 4.2.2.1. Unit Test Code Structure

Classpath to unit test java classes that are in test app in git:

<SmartBookmarkTest>
    app/
        src/test/java/
            com/example/user/activity/
                ClassNameTest

#### 4.2.2.2. Instrumental Test Code Structure

Classpath to instruction test java classes that are in androidTest app in git:

<SmartBookmarkTest>
app/
src/androidTest/java/com/example/user
activity/
activity_ClassName_TEST

### 4.2.3. Server Code Structure

Classpath to server test classes in git:

<ServerTest>
tests
test_views.py

## 4.3. Test Results

### 4.3.1. Bookmark Device Test Results

#### 4.3.1.1. Hardware Components Test Results

The three main components integrated with the Arduino board, which are HC-05 Bluetooth module, LDR (Light sensor), and Potentiometer (Angular Displacement Sensor) were tested based on the plan in Title 4.1.1.1, after required configurations were set in the circuit. All tests passed successfully.

| Component | Expectation | Result |
|---|---|---|
| HC-05 Bluetooth Module | In unpaired case: LED on module is blinking all the time<br>In paired case: LED on the module is blinking once in 2 second | Passed (LED's blinking time changes based on scenarios) |
| LDR (Light Sensor) | Sensor output values written in serial Port change with light conditions | Passed (Sensor output values are written in Serial Monitor and they changed based on light conditions) |
| Potentiometer (Angular Displacement Sensor) | Sensor output values change with angular displacement | Passed (Sensor output values are written in Serial Monitor and they changed based on angular displacement) |

#### 4.3.1.2. Serial Communication Test Scenarios

Serial Communication tests were done through Serial Monitor of Arduino IDE. In each Round- Robin task triggered by Android application, tested values were printed to Serial Monitor.

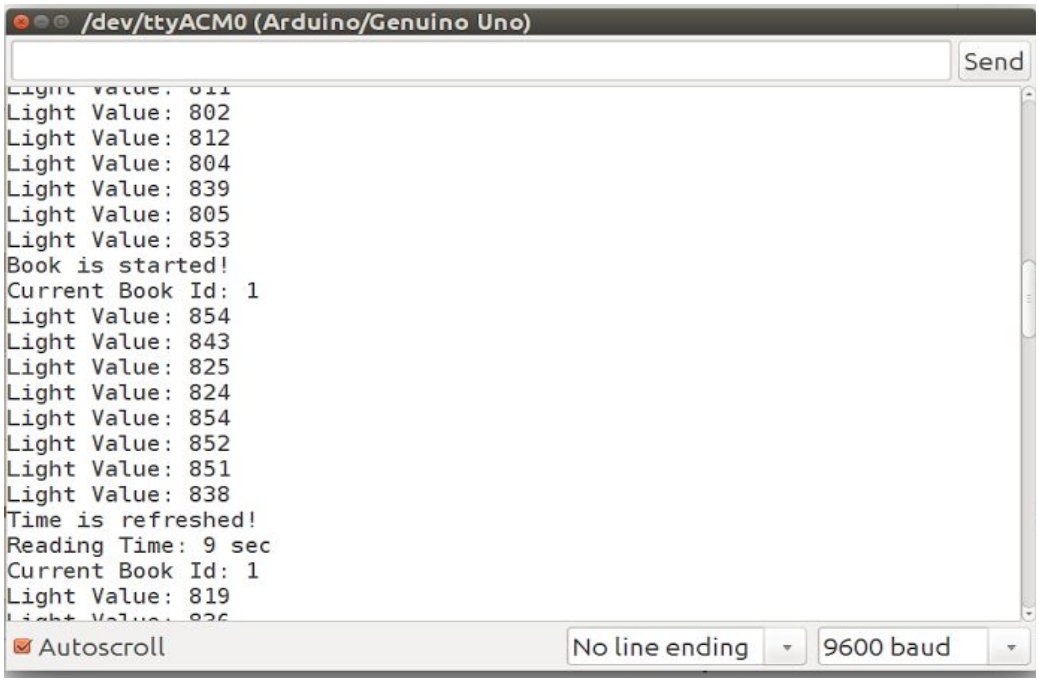


**Figure 6**

| **Scenario** | **Round Robin Task** | **Trigger In Android** | **Result** |
|---|---|---|---|
| Start Book | ListenTask | startButton | Started successfully |
| Calculate Reading/Non-reading Time | LightTask | - (Triggered in Round Robin itself) | Calculated successfully |
| Pause Book | ListenTask | pauseButton | Paused successfully |
| Continue Book | ListenTask | continueButton | Continued successfully |
| Finish Book | ListenTask | finishButton | Finished successfully |
| Calibrate | ListenTask | measureReadingRatio | Calibrated successfully |

| | | | |
|---|---|---|---|
| | PotTask | Button | |
| Send Data | SendTask | refreshButton | Sent successfully |

## 4.3.2. Mobile Application Test Results

### 4.3.2.1. Unit Test Results

| **Test Class** | **Test Function** | **Result** |
|---|---|---|
| AccountHelperTest | getTokenKey_isCorrect | “Correct” |
| AccountHelperTest | getServerAddress_isCorrect | “Correct” |
| BookAdaptorTest | updateStatusTest | “Correct” |
| BookAdaptorTest | updateDropTest | “Correct” |
| BookAdaptorTest | updateIsSyncTest | “Correct” |
| OCRAdaptorTest | updateIsSyncTest | “Correct” |
| ReadingRatioAdapterTest | getStartMeasureTest | “Correct” |
| ReadingTimeAdaptorTest | sumTimeDataTest | “Correct” |

### 4.3.2.2. Instrumental Test Results

| **Scenario** | **Java Class** | **Component** | **Result** |
|---|---|---|---|
| Login | LoginActivity | username,password, loginbutton | Pass |

| | | | |
|---|---|---|---|
| Add Friend | AddFriendActivity | username, searchbutton | Added Succesfully |
| Edit Book | EditBookActivity | booktitle,author,isbn, editbutton | Saved Successfully |
| Add book by Manual ISBN | ManulISBNActivity | bookISBN,Searchbutton | Found Successfully and added<br>Not Found |
| Add Quote Manually | ManualQuoteActivity | quote,page,addbutton | Added Succesfully |
| Pair Device with Mobile Phone | PairDeviceActivity | pairbutton | Paired |
| User Profile Edit | ProfileActivity | bio,email,birthdate,savebutton | Saved Successfully |
| Recommendations | RecommendationActivity | listview | Return Recommendations |
| Scan Book ISBN Barcode | ScanBarcodeActivity | Surfaceview | Scanned Successfully |
| Settings | SettingsActivity | adddevice,changedevice ,notificationsettings | Changed and saved Successfully |
| Friends | Fragment Friends | Listview, username | View Succesfully |
| Add Book | Fragment Add | buttonmanual,buttonscan,buttonrecommendations | Passed Successfully |
| Current | Fragment Current | bookname,bookISBN,bookauthour,progr | Passed Successfully |

| | | | |
|---|---|---|---|
| | | essbar | |
| Bookshelf | Fragment Wishlist | Listview, Startbutton, continuebutton | Passed Successfully |
| Finish Books | Fragment Finish | Listview,statisticsbutton | Passed Successfully |

### 4.3.3. Server Test Results

| **Scenario** | **Test Functions** | **Results** |
|---|---|---|
| Registration | registration | User type json (rest-auth/registration/) |
| Login | login | User type json (account/) |
| Profile Details | profile_details | Profile type json (account/profile/details) |
| Profile Edit | profile_edit | Profile type json (account/profile/edit) |
| Add Book | add_book | Book type json (account/books/add) |
| All Books | all_books | Book type json (account/books/all) |
| Categorized Books | categorized_books | Book type json (account/books/categorized) |
| Book Details | book_details | Book type json (account/books/show/details) |
| Update Book | update_book | Book type json (account/books/update) |
| Sync Book | sync_book | Book type json (account/books/sync) |

| | | |
|---|---|---|
| Delete Book | delete_book | Book type json (account/books/delete) |
| Rate Book | rate_book | Book type json (account/books/rate) |
| Successful Registration | successful_registration | User type json |
| Unsuccessful Registration | unsuccessful_registration | User type json |
| Successful Login | successful_login | User type json |
| Unsuccessful Login | unsuccessful_login | User type json |
| Successful Profile Edit | successful_profile_edit | Profile type json |
| Successful Profile Details | successful_profile_details | Profile type json |
| Successful Add Book | successful_add_book | Book type json |
| Unsuccessful Add Book | unsuccessful_add_book | Book type json |
| Successful All Books | successful_all_books | Book type json |
| Successful Categorized Books | successful_categorized_books | Book type json |
| Successful Book Details | successful_book_details | Book type json |
| Successful Update Book | successful_update_book | Book type json |
| Successful Sync Books | successful_sync_books | Book type json |
| Successful Delete Book | successful_delete_book | Book type json |
| Successful Rate Book | successful_rate_book | Book type json |