PROJECT PLAN

Individual Assignment

by Lachezar Mitov

Contents

Document versioning

Version	Description	Date
1.0	- Added structure of the document - Added main paragraphs - Added delivery estimations	07-09-2022
1.1	- Added Problem Description and Deliverables	08-09-2022
1.2	- Added Use Cases, Functional Requirements and URL Design	09-09-2022
1.3	- Updated the PUT operations in URL Design - Add Test plan Summary and Estimations in User stories	14-09-2022
1.4	- Added link to Test Plan - Rearranged User Stories and added new ones	27-11-2022

Introduction

This is the Project Management Plan for the Individual Assignment. The final product will be a web application. Moreover, the website will allow customers to buy and manage tickets for football matches.

Finally, this document will present the software design of the project and functionalities with their expected deadline.

Problem Description

The solution to the individual assignment will be a software, which will serve as a place for the football fans to buy tickets for matches they want to attend.

For the staff, the software should be able to perform CRUD operations (Create, Add, Update and Delete) for football matches and their respectable tickets.

Moreover, information about the tickets of a football match like participants, date, quantity and price could be added to the software and kept in a database.

Users of the web application are expected to see information about the tickets they have purchased.

Deliverables

Listed below are the deliverables expected at the end of the Sprint 6:

- Database to store all information about the matches and their tickets (date and time, participants, ticket quantity and price)
- The possibility for employees to perform CRUD operations to football matches and tickets
- Website application for the users to see available tickets for future matches and the possibility to buy a ticket(s) if they want to attend a match
- Documentation

User Stories and Acceptance criteria

The user stories are used to describe the functionalities of the software solution in a way that the broad audience can understand it without the need to study the structure of the project. Moreover, the user stories are ordered by their expected delivery deadline and the estimation, based on the Fibonacci Sequence, is used to describe the complexity of each one of them.

- **US01**: As a customer I want to create account, so that my information and purchased tickets are saved
  - **Acceptance criteria**:
    1. The system maintains registering of a user
    2. The system has validation, for the user to create a valid account
    3. The system prevents from created an account with already used email address
  - **Estimation**: 5
  - **Deadline**: Sprint 1
- **US02**: As a customer I want to be able to see all available matches, so that I know which football match ticket to buy
  - **Acceptance criteria**:
    1. The available tickets are displayed.
    2. Tickets for matches with past date are not sold
  - **Estimation**: 2
  - **Deadline**: Sprint 1

- **US03**: As a customer I want to be able to login, so that I can view my purchased tickets and personal information
  - **Acceptance criteria**:
    1. The system maintains authentication & authorization
    2. The system checks for valid email and password
    3. The system prevents unregistered users to login
  - **Estimation**: 3
  - **Deadline**: Sprint 1
- **US04**: As a customer I want to buy a ticket, so that I can watch my favorite football team live
  - **Acceptance criteria**:
    1. The ticket is available and not sold out.
    2. The correct sum of the product is displayed.
    3. The ticket cannot be sold to unauthenticated user.
  - **Estimation**: 5
  - **Deadline**: Sprint 2
- **US05**: As an administrator I want to add matches, so that tickets can be sold for this match
  - **Acceptance criteria**:
    1. The system maintains addition of matches
    2. The match is added with the full and correct information
    3. The system prevents from adding the same match more than once
  - **Estimation**: 8
  - **Deadline**: Sprint 2

- **US06**: As a customer I want to be able to see all the important information about a match before purchase, so that I do not regret my decision
  - **Acceptance criteria**:
    1. The match information is displayed.
  - **Estimation**: 1
  - **Deadline**: Sprint 2
- **US07**: As a customer I want to see my purchased tickets, so that I know which football matches to attend
  - **Acceptance criteria**:
    1. The user is authenticated.
    2. The user has tickets that are purchased.
    3. No tickets will be displayed if the user has not bought any.
  - **Estimation**: 5
  - **Deadline**: Sprint 3
- **US08**: As a customer I want to be able to modify my profile information, so that I keep my information up to date
  - **Acceptance criteria**:
    1. The user is authenticated.
    2. The user’s information is displayed
    3. System supports CRUD operations on users
  - **Estimation**: 5
  - **Deadline**: Sprint 3

- **US09**: As an administrator I want to delete matches, so that customers do not buy tickets about a past event
    - **Acceptance criteria**:
        1. The system maintains deletion of matches
    - **Estimation**: 3
    - **Deadline**: Sprint 3
- **US10**: As an administrator I want to perform CRUD operation on team entities, so that the information is always correct
    - **Acceptance criteria**:
        1. The system maintains CRUD operations of football teams
    - **Estimation**: 5
    - **Deadline**: Sprint 4
- **US11**: As an administrator I want to give accounts administrative permissions, so that new employees can work as expected
    - **Acceptance criteria**:
        1. The system maintains updating user accounts
        2. The system maintains having multiple administrators
    - **Estimation**: 3
    - **Deadline**: Sprint 4

- **US12**: As a customer I want to be able to filter the matches, so that I save time and buy a ticket for the one I want
  - **Acceptance criteria**:
    1. The available matches are displayed.
    2. The filtering system is used correctly.
  - **Estimation**: 8
  - **Deadline**: Sprint 5
- **US13**: As an administrator I want to be able to modify existing matches, so that the information displayed is always correct
  - **Acceptance criteria**:
    1. There are matches that are displayed.
    2. The software maintains the modification of matches
    3. The software has validation on input for matches
  - **Estimation**: 5
  - **Deadline**: Sprint 5
- **US14**: As an administrator I want to be able to see basic statistics about ticket sales, so that the business process can be improved
  - **Acceptance criteria**:
    1. Statistics about sales are maintained by the system.
    2. Statistics about sales are displayed by the system.
  - **Estimation**: 8
  - **Deadline**: Sprint 6

Test plan Summary

The software solution will be tested using Unit Testing for testing the Business Logic Layer of the software. Moreover, Integrational Testing will be used to test the front-end part of the project. All of this is made to ensure that all setbacks in the development stage are tackled and eliminated so that the software works as it is expected.

For more information about testing please refer to \this document : Test Plan

Functional Requirements

Employee side of application

Features include:

- **FR-01**: Staff should be able to create and delete matches
- **FR-02**: Staff should be able to read and update match information
- **FR-03**: Staff should be able to see sales statistics about tickets
- **FR-04A**: Authentication
- **FR-04B**: Authorization

Client side of application

Features include:

- **FR-04A**: Authentication
- **FR-05A**: System should be able to show match information
- **FR-05B**: System should be able to show tickets information
- **FR-06**: System should support buying tickets by authenticated users
- **FR-07**: System should support showing authenticated users purchased tickets
- **FR-08**: System should support filtering of matches on different options

Use Cases

UC-1: Staff wants to sign-in

- **Actor:** Staff
- **Trigger:** Load web application
- **Main Success Scenario:**
  1. Navigate to 'Login' page
  2. System requires for username and password
  3. Actor enters their username and password
  4. Actor confirms the action
  5. System approves the credentials, and the user is authenticated
- **Extensions:**

  3a: The provided information is insufficient
  .1: System highlights the fields with insufficient information
  .2: System displays an error message
  .3: Return to MSS step 1

  3b: The provided information is incorrect
  .1: System displays error message
  .2: Return to MSS step 1

UC-2: User wants to sign-out

- **Actor:** Staff/User
- **Trigger:** Pressing the 'Sign Out' button
- **Pre-condition:** Actor should be logged in
- **Main Success Scenario:**
  1. Actor presses the 'Sign out' button.
  2. System terminates actor's session.
  3. Return to UC-1
- **Extensions:**

UC-3: Staff wants to add a match

- **Actor:** Staff
- **Trigger:** Pressing the ‘Add a match’ button
- **Pre-condition:** Actor should be logged in
- **Main Success Scenario:**
    1. Actor navigates to the form for adding a match
    2. Actor enters the necessary information
    3. Actor confirms the action
    4. System adds a match successfully
- **Extensions:**

3a: The provided information is insufficient
.1: System displays error message
.2: Return to MSS step 2
3b: The provided information is invalid
.1: System displays error message
.2: Return to MSS step 2
3c: The match has already been added
.1: System displays error message
.2: Return to MSS step 2

UC-4: Staff wants to delete a match

- **Actor:** Staff
- **Trigger:** Pressing the ‘Delete match’ button
- **Pre-condition:** The match should exist
- **Main Success Scenario:**
    1. Actor navigates to Matches page
    2. Actor chooses the match to be deleted
    3. Actor confirms the action
    4. System performs the action and notifies the user
- **Extensions:**

2a: Actor has not specified the match
.1: Systems notifies the user to specify the correct match
.2: Return to MSS step 2

UC-5: Staff wants to update a match information

- **Actor:** Staff
- **Trigger:** Pressing the ‘Update match’ button
- **Pre-condition:** The match should be available
- **Main Success Scenario:**
  1. Actor navigates to Matches page
  2. Actor chooses the match to be updated
  3. Actor makes the changes
  4. Actor confirms the action
  5. System performs the action and notifies the user
- **Extensions:**

  2a: Actor has not specified the match

  .1: Systems notifies the user to specify the correct match

  .2: Return to MSS step 2

  5b: Actor has filled invalid data

  .1: Systems displays error message

  .2: Return to MSS step 2

UC-6: Sales manager wants to see ticket sales statistics about a match

- **Actor:** Sales manager
- **Trigger:** Pressing the ‘View Statistics’ button
- **Pre-condition:** Match should be available
- **Main Success Scenario:**
  1. Actor navigates to Match page
  2. Actor chooses a match
  3. System displays the sales statistics
- **Extensions:**

  3a: Actor chooses a match has not sold any tickets

  .1: System notifies the actor that no data is available

  .2: Return to MSS step 1

  3a: Actor chooses a match, which tickets are still not released

  .1: System display error message

  .2: Return to MSS step 1

UC-7: User want to buy a ticket for a match

- **Actor:** User
- **Trigger:** Pressing the ‘Buy ticket’ button
- **Pre-condition:** Match should be available
- **Main Success Scenario:**
    1. Actor navigates to Matches page
    2. Actor chooses the match to buy ticket for
    3. Actor confirms the action
    4. Actor receives his ticket
- **Extensions:**

    2a: Actor is not logged in

    .1: System redirects actor to Login page

    .2: Return MSS step 1

    2b: All the tickets for the match have been sold

    .1: Actor cannot buy a ticket

    .2: System notifies the user that purchasing a ticket is not possible

    .3: Return MSS step 1

UC-8: User wants to see available matches

- **Actor:** User
- **Trigger:** Pressing the ‘Matches’ button
- **Pre-condition:** A match should be available
- **Main Success Scenario:**
    1. Actor navigates to the ‘Matches’ page

    Extensions:

    1a: ‘Matches’ page is empty

    .1: Page is empty

    .2: Return MSS step 1

UC-9: User wants to see ticket information about a match

- **Actor:** User
- **Trigger:** Pressing the ‘Buy tickets’ button
- **Pre-condition:** Match should be available
- **Main Success Scenario:**
  1. Actor navigates to the ‘Matches’ page
  2. Actor chooses a match and presses the button
- **Extensions:**

  1a: ‘Matches’ page is empty
  .1: Page is empty
  .2: Return MSS step 1
  2a: Tickets are still unavailable
  .1: Button is locked
  .2: Return MSS step 1

UC-10: User wants to search for a match using filter

- **Actor:** User
- **Trigger:** Pressing the ‘Filter’ button
- **Pre-condition:** Match should be available
- **Main Success Scenario:**
  1. Actor navigates to the ‘Matches’ page
  2. Actor chooses an option in the filter menu
  3. The match(es) containing this filter are shown
- **Extensions:**

  1a: ‘Matches’ page is empty
  .1: Page is empty
  .2: Return MSS step 1
  2a: No available matches with this filter
  .1: System notifies the user
  .2: Return MSS step 1

UC-11: User wants to create an account

- **Actor:** User
- **Trigger:** Pressing the 'Register' button
- **Pre-condition:** Web application should be open
- **Main Success Scenario:**
    1. Actor navigates to the 'Register' page
    2. Actor fills the form
    3. Actor confirms the action
    4. System creates an account
- **Extensions:**

    3a: The provided information is incorrect
    .1: Systems displays error message
    .2: Return MSS step 2
    3b: The provided credentials are already used
    .1: Systems displays error message
    .2: Return MSS step 2

UC-12: User wants to see their purchased tickets

- **Actor:** User
- **Trigger:** Pressing the 'Log In' button
- **Pre-condition:** Actor should be logged in
- **Main Success Scenario:**
    1. Actor navigates to the 'Login' page
    2. Actor logs in with their credentials
    3. Actor is navigated to the profile page
    4. Purchased tickets are displayed
- **Extensions:**

    2a: The provided credentials are incorrect
    .1: Systems displays error message
    .2: Return MSS step 2
    2b: Actor has not purchased any tickets
    .1: Purchased tickets section is empty
    .2: End of use case

Design

URL Design

Below in the table are listed the URLs that will be used to service the software solution.

URL	Resource	Operation	
/matches/	Matches	GET	Read a list with all available matches
/matches/123	Matches	GET	Read the match with ID 123
/matches/456	Matches	DELETE	Delete the match with ID 456
/matches/789	Matches	PUT	Update match with ID 789
/matches/	Matches	POST	Add a new match
/tickets/	Tickets	GET	Read all tickets in the system
/tickets/123	Tickets	GET	Read the ticket with ID 123
/users/	Users	GET	Real all users from the system
/users/123	Users	GET	Read the user with ID 123
/users/456	Users	DELETE	Delete the user with ID 456
/users/789	Users	PUT	Update user with ID 789
/users/	Users	POST	Create new user