Project Plan

TSH Event Management Tool

TSH

Date	**:**	**22-09-2023**
Version	**:**	**1.1.0**
State	**:**	**In progress**
Author	**:**	**Aleksandar Popov**

Version history

Version	Date	Author(s)	Changes	State
0.1	18-09-2023	Aleksandar Popov	Initial document	Draft

0.2	19-09-2023	Aleksandar Popov	Glossary, project definition, phasing, moscow list	Draft
0.3	21.09.2023	Aleksandar Popov	Styling, mistakes, feedback	Draft
1.0	26.09.2023	Aleksandar Popov	Contents, table	Finished

Contents

1. Glossary

- TSH – The Student Hotel / The Social Hub
- UI – User interface
- UX – User experience
- URS – User Requirements Specification

2. Project assignment

2.1 Context

The Social Hub or as it is formerly known as The Student Hotel is a Dutch company with branches all over the world. It welcomes students from all over the world and strives to accommodate them in a welcoming and social environment. That is why every month events are organized to entertain and help these students meet new people.

2.2 Project definition

As a student living in TSH and helping the event organizer with the planning I could not help but notice that the structure of the planning of events and organization was a bit chaotic and not fully efficient. The current methods of organizing and promoting the events are not always optimal and lead to fewer people attending said events.

2.3 Goal of the project

The solution aims to facilitate the flow of information and put everything related to the students and events in one place. Means of communication and promotion of events are scattered and not always efficient, that is why emails and push notifications sent from the application are more suitable. The only way to receive any pictures taken during events currently is by contacting the event organizer, which is not always possible, that is why uploading images will make it easy for students to receive them. The desktop app will serve as an administration portal for managing users and branches and the web application will take the responsibility for event management and will be the interface for users to interact with the system.

2.4 Strategy

The methodology used will be agile. As can be seen from the phasing plan below, the project will be implemented by one-week sprints at the end of which should be a semi-production ready and functional application. The time periods (divided by weeks) are designed to provide enough time for implementation, testing and ensure that the feature/change is ready before the end of the sprint and will not have to continue during the next week.

3. Project organization

3.1 Stakeholders and team members

- **Client**

  TSH is an international student housing and hotel company with headquarters in Amsterdam. It provides both temporary accommodation and long-term stay for students and tourists.

- **Team**

The only developer working on this project would be me. I also happen to be a client since I continuously help with the organization of the events of the TSH as I live there.

Name	Abbreviation	Role and functions	Availability
Aleksandar Popov *aleksandar.popov@student.fontys.nl*	*Developer*	*Developer*	*Always*
Jessie Oirschot	*Event organizer – Eindhoven*	Stakeholder *Event organizer in the Eindhoven branch*	*Once a month during workdays*

3.2 Communication

Meetings with teachers should occur at least once every week.
Meetings with the client should occur at most once a month.

4. Activities and time plan

4.1 Phasing

Week	Work
Analysis	
1	• Ideation document • start URS
2	• Project plan • Finish URS
3	• Test case • Implement feedback
Design	
4	• Database diagram

	• UML
Implementation	
5	• Reasearch encryption • Improve current encryption • Improve other security features (cookies...)
6	• Refactor according to the SOLID principles • Implement clean code practices
7	• Unit tests • Event filtering
8	• Implement image-sharing in Logic and data layers
9	• Implement UI interface for image-sharing
10	• Research push notifications and emails
11	• Implement push notifications and emails
12	• Discount codes • Improve UI styling
Testing	
13	• Manual testing • Bug fixing
14	• Unit testing • Bug fixing
15	• Final additions and changes
16	• Presentation

3.2 MoSCoW:

- Must have:
    - Users and content must be divided into branches.
    - List of the events for the month.
    - Sign up and cancel for an event as a student.
    - Create events as an organizer.
    - Statistics for the events.
- Should have:
    - Calendar overview of the events.
    - Place to share images.
    - PWA
- Could have:
    - Notifications.
    - Ticket System with QR codes.
    - Import users from .csv
    - Email sending service.
- Will not have:
    - Integration with existing database/website of the company.

5. Testing strategy and configuration management

5.1 Testing strategy

The tested components will be the Logic layer and the SQL query builder. The other parts of the project are not going to be tested, since testing them would mean that the technologies used for them are tested, which goes against the TDD (Test Driven Development) Principle of not testing third party code. The tests will follow the AAA strategy – Arrange, Act, Assert.

Manual testing will be implied while working on the project as well as final testing during the final weeks. Testing will be done from the following points of view: a developer, student, event organizer and a user who has no context of the application and no knowledge in programming or any specific field. The first one will be inevitably used especially during implementation, the second and third ones are the actual target users, and their points of view are meant to improve the UX and fit the requirements for the application. The last one is the lowest level of understanding, which will ensure that any person can make use and sense of the UI and that it is not overcomplicated.

5.2 Test environment and required resources.

The framework used for testing is MSTest. No database is required since the Data layer is mocked. Every tested entity will have at least one or two default entities in the mock Data layer.

5.3 Configuration management

- VCS – git
- Branching method – per feature
- Commit messages - under 50 characters if possible.
- Platform – Gitlab
- Continuation – git fork

6. Finances and risk

6.1 Project budget and constraints

- Time – from 05.09.2023 to 12.01.2024 (16 weeks)
- URS
- Technologies:
  - .NET – Razor pages
  - WinForms
  - MySQL/MSSQL
  - Luna server

6.2 Risk and mitigation

	Risk	Prevention activities	Impact	Chance of occurrence
1	Not meeting deadlines	Follow phasing plan	Moderate	Moderate
2	Not developing product according to the client expectations	Follow URS and meet with client	Low	Low
3	Deliver unmaintainable, not fully functional product	Meet and discuss with teachers and follow best-practices and principles	Heavy	Low