Testing my Bachelor Thesis’s application

Testing is among the most important aspects of software development. It involves the process of running a program with the aim of detecting errors and ensuring that the software meets the specified requirements. Typically, testing is performed after the development phase and before the software is released. By testing the application, developers can identify bugs, errors, and defects, and verify that the application performs as intended. Another significant aspect is that testing is critical in ensuring that the software is reliable, efficient, and user-friendly. Ultimately, testing helps to enhance the quality of the software, which is vital for establishing trust with users and maintaining a positive reputation in the market.

Almost a year ago, the idea for meeTeam was born. The social networking mobile application focuses on team sports and allows people from different backgrounds and interests to connect based on their favorite sport. Users can create events for specific sports (Soccer, Basketball, Tennis, or Volleyball) and customize them to their liking. For example, they can choose to make the event just for fun or competitive. Other users can then join the event as long as there are still available spots.

One of the app's main features is the ability to book the location for the event directly through the app, without the need to call or contact the location manager. However, the biggest selling point is the introduction of gamification. Users can collect XP, see their progress (games played), and rank up in different leagues based on their performance.

Going back to the development stage of an application, various testing techniques should be used to ensure that the software is of the desired quality. These techniques include unit testing, integration testing, and system testing. Firstly, **unit testing** implies testing individual components or modules of the application to ensure that they function correctly. Moreover, **integration testing** involves testing the interaction between different components of the software to ensure that they work together as expected. On the other hand, **system testing** implies testing the

entire system as a whole to ensure that it meets the specified requirements. Other testing techniques such as acceptance testing, performance testing, security testing, and usability testing may also be used depending on the software's requirements and the development team's needs.

One important aspect of testing the meeTeam application is testing the feature that allows users to book locations for their events. This involves testing the component that handles the booking process to ensure that it functions correctly.

Unit Testing

To start, one possible scenario for unit tests would be validating the **retrieval of the available locations**. This involves creating test cases that simulate different scenarios, such as empty locations, multiple locations, or specific filtering options.

Integration Testing

One crucial integration test can involve verifying **the interaction between the location selection module and the date selection module**. This test ensures that when a user chooses a location, the date selection module accurately updates and displays the available dates specific to that location. By simulating different location and date combinations, you can ensure that the integration between these components is functioning correctly.

System testing

To ensure the reliability and effectiveness of the book location feature, system testing should encompass testing scenarios that cover the entire booking process, from location selection to payment confirmation. This involves creating comprehensive test cases that simulate different user journeys, such as **selecting locations with varying availability, choosing dates with different time slot options**, and **successfully completing the payment process via Apple Pay**.