

SIT725 - Planning to Test (6.1P)

Student Information

- **Student ID:** s217551712
 - **Name:** Antheis Maxenn
 - **Task:** 6.1P - Planning to Test
-

Project Title

EventSpot - Local Event Discovery Platform

Testing Strategy for EventSpot

After reviewing the lecture, I believe a robust testing strategy is essential for EventSpot to ensure students can rely on it for finding local events. We need to move beyond just "checking if it runs" and adopt a structured approach involving verification and validation throughout our agile sprints.

Unit Testing (White-Box)

For **Unit Testing**, which is our white-box testing, I plan to use **Jest** to test individual functions in isolation. Since we are using Node.js, this is critical for our backend logic.

- **Example:** I will write tests for our user authentication helper functions to ensure passwords are hashed correctly, and for our event validation logic to make sure users can't post events with dates in the past.

Integration Testing

Integration Testing is where we will verify that our modules interact correctly. I need to ensure that our Express.js server communicates properly with the MongoDB database.

- **Example:** A key test here would be checking that when a user posts a review, it not only saves to the database but also correctly updates the event's average rating. We also need to test that our Socket.io implementation actually triggers notifications to other clients when an event is updated.

End-to-End (E2E) Testing

Finally, for **End-to-End (E2E) Testing**, we will simulate real user journeys to validate the entire stack.

- **Example:** I'll create test cases that mimic a student signing up, creating a "Study Group" event, and verifying that the pin actually appears on the Google Map. This ensures that our frontend, backend, and external APIs are all working together seamlessly to deliver the right experience.