# **Sprint Review 2**

**Scrum 2:**

Scrum Master: Liam Blair

Product Owner: Adam Green

Development Team: Ashley Drexler, Aaron Schanck, Lakshmana Silva

**Scrum 3:**

Scrum Master: Aaron Schanck

Product Owner: Ashley Drexler

Development Team: Adam Green, Liam Blair

Not Present: Lakshmana Silva

**Features implemented**

* We created Flask API to serve our application. The Flask API connects to the database and responds to HTTP requests made to it. The Flask API was deployed onto our cloud host on the uWSGI software application.
* Began UI development within the application. We plan to further develop our UI in this coming sprint.
  + The login page is fully fleshed out with its buttons, despite none of them having functionalities. We plan to further develop our UI in this coming sprint.
  + A logout button has been added (currently non-functional).
* We were able to connect from the application to the API and received the correct data from the database, thus showcasing the proof of concept for the API endpoints.
* We created the Use Case Model Document.
* The README file has been created.

**Issues fixed**

* We were unable to connect directly to the remote database from Android so we needed to create a web application to serve as an API to make the calls to the database.
* There were a couple errors deploying the web application on the server that we were able to resolve.
* A couple of visual defects in the UI/UX of the Login page were fixed.
  + Issue #35: Login input boxes hide text
  + Issue #36: Password input shows password

**Implementation review**

What Went Well:

1. README file created with zero issue.
2. Created use case documents including descriptions and diagrams successfully and in a timely manner.

|  |  |
| --- | --- |
| **USE CASE DIAGRAMS** | **USE CASE DESCRIPTIONS** |
| 1. Account System 2. Administration System 3. Chat System 4. Payment Methods System 5. Purchase History System 6. Recommendations System 7. Search System 8. Signup System 9. Tracking System 10. Bidding System 11. Item Cart System 12. Item Catalog System 13. Premium System 14. Review System | 1. Add Item 2. Ban Account 3. Cancel Premium Subscription 4. Create Account 5. Edit Account Info 6. Enter Item Page 7. Enter New Payment Method 8. Filter Search 9. Leave Review 10. Login 11. Logout 12. Place Bid 13. Remove Item 14. Remove Product 15. Select Payment Method 16. Send Message 17. Premium System 18. View Item Location 19. View Purchase History 20. View Search |

What Problems Occurred:

1. Initial error connecting to the remote database.
2. Visual defects that popped up while creating the Login page.
3. Errors deploying the web app on server.

How Problems Were Solved:

1. Error connecting to the remote database was solved by incorporating Flask API.
2. Errors deploying the web app was solved by fixing minor bugs.
3. Problems with the visual defects were solved by fixing minor bugs.

**Changes made**

* Changed from a multiple database system to a single database system with multiple tables. This change allows for easier connections from the Flask API and alleviates possible security issues.

**Plans for next sprint** *(What will be done for the next sprint)(Priorities based on 1-5, with 5 being most important)*

* Secure our Flask API using NginX and SSL certificates. Priority: 4
* Secure the request handshake with JWT to allow for authentication between client and server. Priority: 3
* Construct functions to handle API endpoints on the client. These functions should be able to handle the communication from start to finish and return the correct output for the specified action. Priority: 5
* Finish user accounts on the client side. This includes: logging in, forgot your password, logging in as guest, and premium accounts. Priority: 5
* Finalize UI mockup, attempt to make decent ground on the UI design within the application (layout.xml design). Priority: 5
* Construct item table. Priority: 4
* Add models and service for communication in Flask API. Priority: 4
* Construct email notification system/Android notification system. Priority: 2
* Build Logger within Android. Priority: 4
* Begin testing Flask API. Priority: 3

**Scrum Review** *(What went well in Scrum, what could be improved, and what changes will be made)*

***What Went Well:***

* Team had well-defined roles that led to better efficiency.
* Scrum progressed quickly.

***What Could Be Improved:***

* Attendance: Aim for full attendance next scrum.

***Changes To Be Made:***

* *N/A*