

Team 15 Sprint 2 Planning Document
Delivery and Order Management System(DOMS)
Kunal Goyal, Hemil Desai, Shivank Tibrewal, Dinesh Gajwani

Sprint Overview

The goal for our second sprint is to get started with core user stories of our project involving the algorithm and automation of the app. At the end of the sprint, we plan to have a minimum viable product that can be represented with user functionality and automation.

We have described the user stories, we aim to implement in this sprint. Along with each user story, are the key technical features associated with it. Each user story has been assigned to a team member with the number of hours expected to complete it.

Scrum Master: Dinesh Gajwani

Scrum Meeting Time: 1:30pm MWF

Risks/Challenges: Getting familiar with new libraries and frameworks in Android including authentication and proper backend connections to ensure smooth operations.

Current Sprint Detail

User Story 1: Orders can be assigned to available drivers manually.

| Task Description | Estimated Time | Owner |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------|
| Create the data structure and algorithm for delivery scheduling - First Come First Serve. Explore different services like Redis for creating fast reliable queues. | 5 | Hemil |
| Pop the first order and the first available driver, assign the order to the driver and update the database and the frontend. Interact with | 4 | Hemil |

| | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------|
| the data store, update the database and send the data in JSON format. | | |
| See the list of available drivers and pending orders - updated in real time. Use ajax requests to handle real time updates in driver and order status. | 7 | Shivank |

Acceptance Criteria: If this user story is implemented well, the tester should be able to assign drivers to active orders in the system.

User Story 2: As an Admin, I would want to manage store owners (Add, Remove, View) into the system.

| Task Description | Estimated Time | Owner |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------|
| Construct a simple web page using HTML, CSS and Javascript form where the details of a store owner can be added and submitted through an HTML request | 6 | Kunal |
| Perform Validation to check if store owner already exists or user entered invalid data | 2 | Dinesh |
| Construct a simple web page using HTML, CSS and Javascript form where the details of a store owner can be loaded from the database through an HTML request | 5 | Kunal |
| Construct a simple web page using HTML, CSS and Javascript form where all store owners in the database are listed through an HTML request | 5 | Kunal |

Acceptance Criteria: If this user story is implemented well, the tester should be able to add a store owner, view all store owners in the system and delete a store owner.

User Story 3: As an Admin, I would want to manage drivers (Add, Remove, View) into the system.

| Task Description | Estimated Time | Owner |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------|
| Construct a simple web page using HTML, CSS and Javascript form where the details of a driver can be added and submitted through an HTML request | 6 | Kunal |
| Perform Validation to check if driver already exists or user entered invalid data | 2 | Dinesh |
| Construct a simple web page using HTML, CSS and Javascript form where the details of a driver can be loaded from the database through an HTML request | 5 | Kunal |
| Construct a simple web page using HTML, CSS and Javascript form where all drivers in the database are listed through an HTML request | 5 | Kunal |

Acceptance Criteria: If this user story is implemented well, the tester should be able to add a driver, view all drivers in the system and delete a driver.

User Story 4: Automate order assignment to drivers.

| Task Description | Estimated Time | Owner |
|-----------------------------------------------------------------|----------------|-------|
| Create a basic first come, first serve algorithm using a queue. | 6 | Hemil |

| | | |
|---------------------------------------|---|-------|
| Test the algorithm for corner cases. | 2 | Hemil |
| Link the algorithm to the backend API | 6 | Hemil |

Acceptance Criteria: If this user story is implemented well, the tester should be able to efficiently test the algorithm by creating dummy orders using faker and see the scheduling of deliveries in real time on the store dashboard along with the list of drivers and orders being updated in a timely manner.

User Story 5: Setup the mobile app and create the splash pages.

| Task Description | Estimated Time | Owner |
|-----------------------------------------------------------------------------------------------------------------|----------------|---------|
| Research and gain knowledge on XML and Java programming on Android Studio. | 5 | Shivank |
| Research and gain knowledge on creating splash pages on Android Studio. | 5 | Shivank |
| Research, construct and define a simple splash page using XML and Java for the DOMS Application. | 8 | Shivank |
| Research and construct the main application page which the splash page will transition into using XML and Java. | 8 | Dinesh |

Acceptance Criteria: --

User Story 6: Create driver sign up in the Android mobile app.

| Task Description | Estimated Time | Owner |
|------------------------------------------|----------------|--------|
| Research and gain knowledge on HTML post | 5 | Dinesh |

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---|--------|
| request and JSON on Android and decide what to use later in the application. | | |
| Construct a simple sign up page using XML and Java where the details of a driver can be added and submitted through an HTML post request or JSON request. | 8 | Dinesh |
| Perform Validation to check if driver already exists or user entered invalid data using HTML get requests or JSON request. | 5 | Hemil |

Acceptance Criteria:

If this user story is implemented well, the tester should be able to sign up as a driver successfully, and as an admin, be able to view and accept requests.

User Story 7: As an existing driver, I need to log into the DOMS Android Application.

| Task Description | Estimated Time | Owner |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------|
| Construct a simple login page using XML and Java where the details of a driver can be added and submitted through an HTML post request or JSON request. | 6 | Shivank |
| Receive the HTML request and authenticate the user using passportJS. | 2 | Dinesh |
| If valid driver, get driver information using HTML get request or JSON | 5 | Hemil |

| | | |
|--------------------------------------------------------------------------------------------------------------|---|--------|
| request | | |
| Redirect the user to a new logged-in-page using XML and Java where he or she can view their own information. | 5 | Dinesh |

Acceptance Criteria: If this user story is implemented well, the tester should be able to log in to the system if he or she exists in the system and should not be able to log in if he or she doesn't exist.

Remaining Backlog

Functional

1. As an administrator, I should be able to:
 - a. View feedback by users
 - b. Add other administrators
 - c. View scheduled / completed delivery orders
2. As a driver, I should be able to:
 - a. Accept and reject delivery
 - b. Update delivery status
 - c. See the delivery address on the map
 - d. Get the best routes available and an ETA.
3. As a store owner, I should be able to:
 - a. View feedback from users
4. As a customer (TBD), I should be able to:
 - a. Provide feedback for the service.
 - b. Pay tip.
 - c. Get an ETA.
5. As a developer (TBD), I should be able to:
 - a. Use the tools of the platform through an open API
 - b. Use simple HTTP protocols to place orders via the API
 - c. Integrate seamless payments.

d. Get feedback based on the orders.

Non-Functional

1. The application should focus on being user friendly. The application should not have too many buttons on one screen as that can overwhelm the user. Instead, we should make it simple and understandable.
2. The system should have fast response times and efficient latency and bandwidth. The system should have an online response time of no more than 10 seconds.
3. The server must be able to handle concurrent users.
4. The app should aim to have minimum downtime.
5. The API and database should be scalable to handle occasional overloads and several users.
6. The server should be efficient. If deployed on cloud, it should integrate well with services like Amazon AWS or Microsoft Azure.
7. The platform should have a testing mode and a production mode.