

# Project Experience Report

URXperience - Connecting UR World

ENSE 400/477 Winter 2023

Abraham Murgerwa

Dhruv Modi

Dinesh Dalip

Favor Fasunwon

Table Of Contents

<b>Introduction.....</b>	<b>1</b>
Group Members/Roles.....	1
Problem Definition.....	2
The Golden Circle.....	2
Audience.....	2
<b>Project Planning.....</b>	<b>3</b>
Features selected.....	3
Technology research.....	5
Initial timeline.....	5
Division of labor.....	6
<b>Project Execution.....</b>	<b>7</b>
<b>Project Reflection.....</b>	<b>10</b>

### List of Figures

- Figure 1      Feature Priorities vs Complexity.
- Figure 2      Gantt Chart of Project timeline.
- Figure 3      Events Page.
- Figure 4      Chat system.
- Figure 5      Digital Complaint box.

### List of Tables

- Table 1      Members and roles/Team roles.
- Table 2      Features, Features requested by, Priority level and Complexity.
- Table 3      Summary of project reflection.

## Introduction

This report is composed of the overall experience sort throughout the development process of the URXperience application. Table 1 lists each member's independent role as well as their team based role. The group designed, developed, and implemented an application for the University of Regina student housing; this is a digital ecosystem designed to connect students with one another, their school, and to become more familiar with food related landmarks in Regina. This report will address in detail the experiences the Four Codesmen encountered during their 8 month Capstone project.

### Group Members/Roles

Name	Role	Team Role
Dinesh Dalip	Full Stack Developer	Software Architect
Favor Fasunwon	Full Stack Developer	Team Lead
Dhruv Modi	Full Stack Developer	UI/UX Designer
Abraham Mugerwa	Full Stack Developer	Documentation Manager

*Table 1: Members and roles/Team roles*

### Problem Definition

About 11% of the students at the University of Regina live in residence. However, living on campus does not exclude one from feeling social isolation or lackluster of the full university experience. Thus, a solution was brought to light to revolutionize the way students in residence connected with each other. URXperience is big step into tackling this problem as it allows students to chat with each other, make complaints and stay up to date on occurring events.

### The Golden Circle

**Why** - The purpose of our application is to integrate a digital ecosystem for the student living in residence which evolves the experience of living on campus and makes living on campus more lively.

**How** - The key features of the application are a chat system, events, complaint box, recipes, and take outs which capture all the experiences that a resident on campus that may be new to Regina may seek to enhance.

**What** - Building a web application to help make connecting University Of Regina Residents with each other and their campus easier.

### Audience

The target customers for URXperience are University Of Regina residents. Being that this application is intended to connect those living on campus, including those who are new to Regina, the application intends to encompass their experiences living on campus and increase socialization among their fellow residents. This application, however, was also able to include residence assistants(RA's) who are also students, but are the first point of contact for residents with regards to questions or concerns about their residence experience; This is why it was necessary for the application to include them as they play an integral role in campus life.

## Project Planning

### Features selected

Upon conducting community research to determine important features for our customers, the team identified a list of features and functionality requirements which was consistent throughout the research ; The common features are listed below:

- Login and Registration - to restrict the application to residents at the University Of Regina.
- Chat System - to meet and talk to fellow residence students on campus.
- Complaint box - to address issues of faults on and around residence that require addressing, rather than filling out a form.
- Admin routes - to have more control and responsibility of the application for residence assistants.
- Takeout page - For new residents to Regina to see what food is available to them around them.
- Recipe Page - to give them some ideas on things to cook or learn to cook
- Events Page - to help keep residents up to date on events going on around campus

Table 2 below lists each feature requested by particular members of UofR Residence, how important that feature is to them, and from an implementation standpoint, the complexity for the group to develop the given feature.

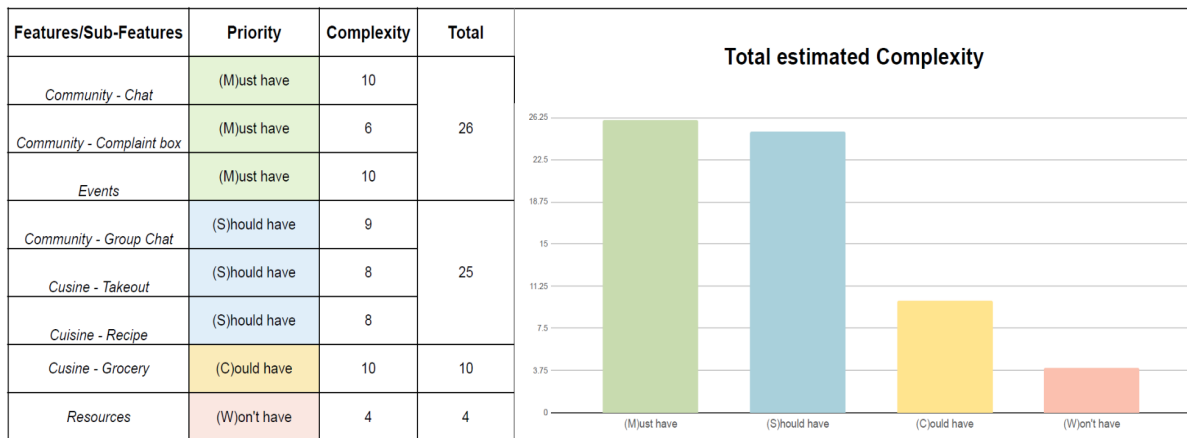
Feature/Sub-Feature	Requested by	Priority(low/medium/high)	Complexity
<b>Login/Registration</b>	RA Manager	High	High

<b>Community/Chat System</b>	UR Residents	High	High
<b>Complaint box</b>	UR Residents/RA	High	Low
<b>Admin routes</b>	RA Manager	High	Medium
<b>Takeouts in the area</b>	UR Residents	Medium	High
<b>Cooking ideas/help</b>	UR Residents	Medium	High
<b>Events</b>	UR Residents	High	High
<b>Security</b>	RA Manager	High	Medium

*Table 2: Features, Features requested by, Priority level and Complexity*

Following the feature selection, the next step for the team was to decide which features to prioritize based on the time and complexity of the envisioned feature. The figure below is a priorities vs complexity graph which is used to better illustrate how the group selected which features to implement first. All the features are listed on the first column, the second column contains the level of priority which was captured in the user research in the previous figure. The priorities are Must have, should have, could have and won't have. From there the group ranked the most complex features to implement based on capability and amount of functionality required. The highest score determines the most complex features and determines which features to begin working on first.

[URXperience Feature Priorities vs Complexity Graph](#)



*Figure 1: Feature Priorities vs Complexity*

### Technology research

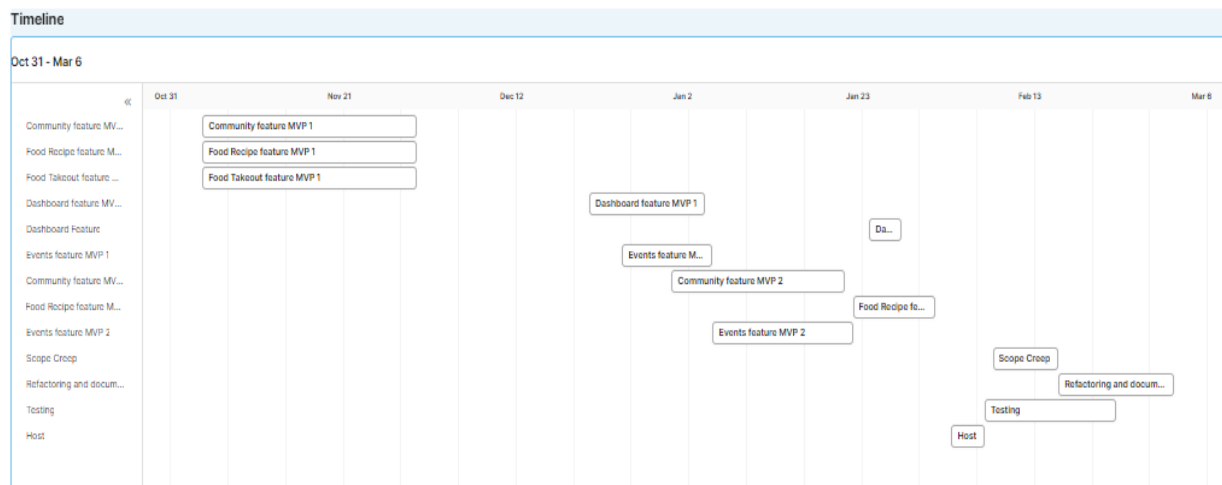
The top three technology stacks that were researched based on the project were two web applications and one mobile application. MERN Stack(a web application), which is MongoDB, ExpressJS, ReactJS, and NodeJS. FARM stack(Web Application), which is FastAPI, React, and MongoDB. Flutter(mobile application) was the third choice. Ultimately the choice was to use the MERN stack after weighing the benefits, which included one or two members with some more understanding of this tech stack versus the other two. MERN stack also had a lot of resources as it was a very popular stack, which allowed the team to watch some tutorials online for free, thus gaining some understanding of the stack.

### Initial timeline

With respect to the development stage of the project, the timeline in figure 2 below shows the three features that were to be started and completed for the first MVP between October and the beginning of December. Upon completion of MVP1, MVP2 started towards the end of January, and went until mid January. This allotted the remainder of time to fix any bugs and refine the



look of the application with some styling following feedback. The month of february was to be dedicated to hosting the platform online and going through some testing(including user testing) along with refactoring. The month of march was dedicated to closing documentation and any last minute changes to the website.



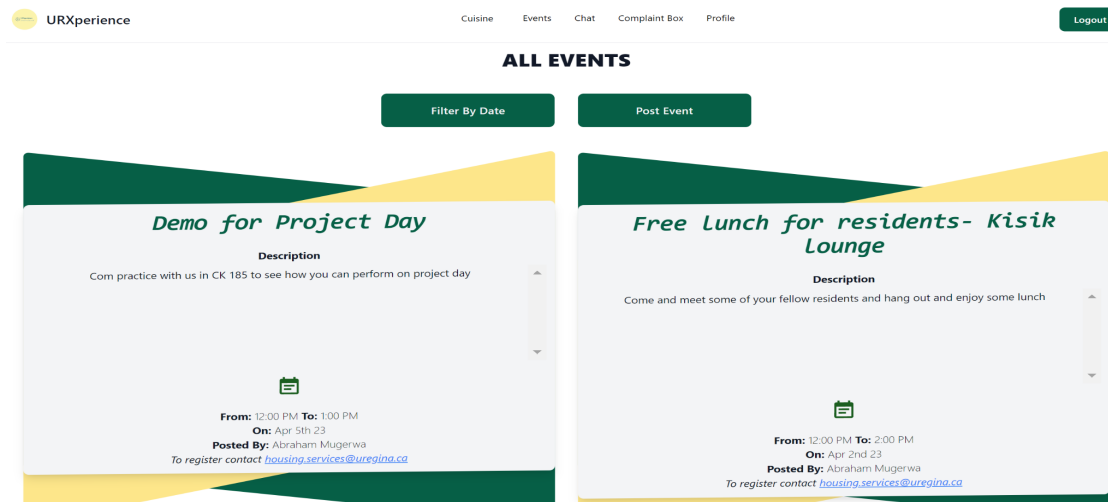
*Figure 2: Gantt Chart of Project timeline*

### Division of labor

Initially the Four Codesmen consisted of two front end developers and two backend developers. Teams were split into two teams(one frontend and one backend) for the first half of development. This allowed for simultaneous development of the features for the Chat system and recipe/takeout feature to be completed in time for MVP 1. Once the teams moved to the latter MVPs, all members were required to be full stack developers in order to complete the remaining MVPs in time. This was found to be efficient and beneficial for the remaining four months of the project because of varying schedules, as well as the frontend and backend knowledge being required to successfully complete the remaining features; This allowed each member to develop skills in areas where they may have previously lacked.

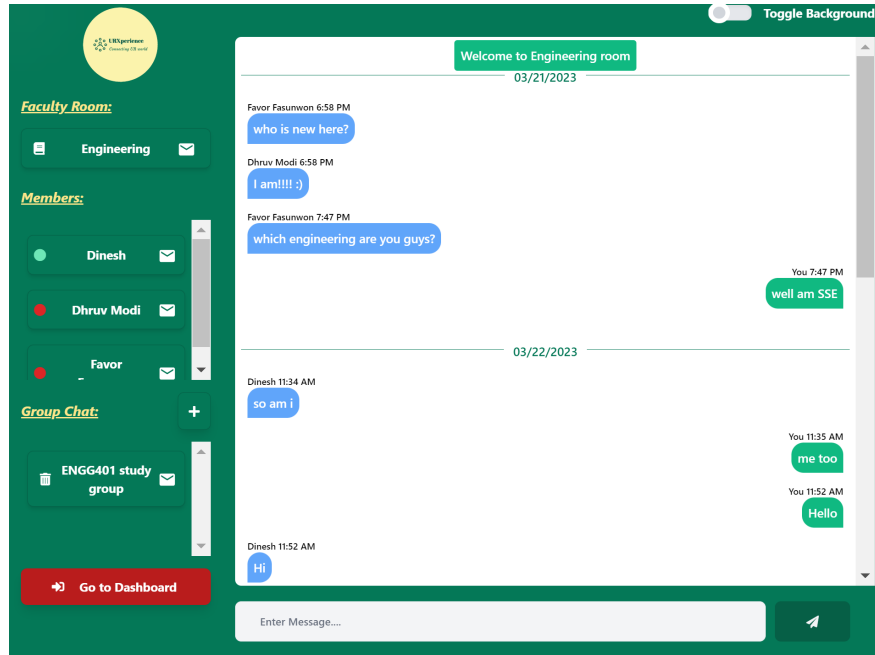
## Project Execution

Figure 3 shows the final version of the Events page, which shows all the upcoming events that are posted by the residence assistants. The RA's are the only users allowed to post events and see the post event button, as it does not show on the students side.



*Figure 3: Events page from RA's side*

The chat feature of the application is where students will get to meet and interact with each other, so it is packed with a few features within. The group decided to admit students into a main faculty room which you are enrolled in. Figure 4 shows a user in the Engineering room, where they can chat to only those on campus who are also in the engineering room. To still allow connection with those outside the students faculty, the members will show all members on residence that can be private messaged. Additionally there is a feature allowing users to make group chats to further their needs, for example, to form study groups.



*Figure 4: Chat Feature*

The digital complaint box is a feature added after some more user feedback was done. Currently the students have to go downstairs and fill out a form at the residence office if there are any complaints in regards to minor things not working or other small non maintenance concerns. This feature allows students to post those issues and submit them without having to go downstairs, eliminating the hassle and the waste. Their complaint is submitted directly to the RA's who will have the option to resolve the issue once it has been resolved. Figure 5 below shows the interface of the complaint feature. The left panel is where complaints can be submitted along with a photo, if applicable. On the right hand side the student will see all their unresolved complaints that they made. Once a complaint is resolved, an automated message is sent to the student to acknowledge this has taken place.

Complaints can be filed using this form and a record of your complaints will be kept in the container on the right (or below for cellphones)

**File your Complaint**

Favor Fasunwon

Location (e.g Kisik Lounge FL2)

Choose File | No file chosen

Please upload an image if applicable

Detailed Description....

Please review your complaint before submitting

**Submit**

**Live, learn. UR HOME.**

**Paskaw towers fl5**

Broken table here at the lounge plz come fix. Thnx

Submission Date: 03/24/2023

**Live, learn. UR HOME.**

**Kisik tower fl2**

Broken table here in need of fixing

Submission Date: 03/24/2023

*Figure 5: Digital Complaint box*

The remaining features that were completed are the foods page which is able to generate different recipes for students to select to learn how to make. Students can choose to sort recipes based on their diet, such as meat lovers, vegetarian, or Vegan. Students can also search for recipes based on items they may have in their fridge to help them see what they can make with ingredients they have. If students are not up for cooking, they can navigate to the takeouts page which is able to show a map of restaurants around their location, or they can search an area around the city. Once they choose a location, the application will show details of the restaurant, with an option to navigate to the website or call the restaurant. Additionally, Students can search for restaurants based on their rating from 3.0 and above, 4.0 and above, or 4.5 and above to show them more varying levels of rated restaurants.

URXperience captures the experiences sought out by students, as the main features requested were all implemented into the application, and the application is available to use. Thanks to the

dedication of each team member, all the planned project activities were met, and the project was completed on time.

### Project Reflection

What went well	<p>The distribution of tasks among group members allowed for independence and skill development. This also allowed for team collaboration when combining tasks together which, when time permitted, was effective in making sure that collaborative features were completed at the right stage and time. Each of the members had the opportunity to work in different pairs when schedules aligned which helped grow the overall team collaboration. Team Collaboration was successful not just from a coding aspect, but also in terms of being on the same page in fulfilling the purpose of the application. The team's management of responsibilities allowed the application to meet the needs of the users.</p>
What didn't go as well	<p>From a technical side, the codebase could have been optimized to better codebase standards to keep things DRY as it would have improved the refactoring and testing of the code. Another technical aspect that did not go as well was hosting the project on a live service as it proved to present more challenges. From a non-technical side, there could have been some improvement</p>

	<p>towards handling new ideas from group members, as it would sometimes conflict with the project timeline and importance to the overall goal of the project. Additionally having more fixed times to complete a feature could have been improved for some of the features that were implemented as some loose dates would sometimes delay the progress anticipated at that point in time.</p>
What was learnt	<p>The importance of user testing and discoverability feedback proved to be valuable as the team was able to make some key changes before completing the project that made a big difference in the final application. Keeping good practice of documenting continuously and as concisely as possible for review and additions is always good practice. With respect to the technology stack, the team got the opportunity to learn/enhance their skills with the MERN stack during this project. Weekly scrum presentations/vlogs, team meeting and discussion skills, and meetings with a mentor were all important to enhancing soft skills and communication. On the other side the team learnt the impact of decision making without all members being present, which on some occasions took place.</p>
What could be done	<p>User research, team decisions with everyone present and in</p>

<p>differently in hindsight</p>	<p>agreement, additional user testing to gain more knowledge, more accurate documentation in planning phase to improve development phase</p>
<p>Project lessons/takeaways for the future</p>	<p>Testing and documenting upon implementing (fast feedback cycle) a new function would be a future lesson to incorporate to future projects, as it will prove to save a lot of time in the end.</p> <p>With learning a new tech stack, and going through Software development life cycle, the team increased their development as future Software Engineers that can provide them with the experience needed to take the next step in their future careers.</p> <p>Soft skills like communication are important in any industry/profession and it was one of the key takeaways from this project that will be invaluable to each member in the future. Lastly, Lifelong relationships were gained from collaborating on this project that will always be cherished.</p>

*Table 3: Summary of project reflection*