Project Proposal

UR QnA

A convenient, fast, and efficient way to answer questions about the University of Regina.

Date/Version: 2021.09.18

Authors: Zakiyyah Noorally, Bernadette Veninata

Course: CS 476 Software Development Project

Table of Contents

Purpose of This Document

This document will provide an understanding of the project undertaken by the authors to fulfill the course requirements of their software development course (CS 476). It also serves as a baseline for project planning from start to successful closure. Project objectives, scope, requirements, expected schedule and budget are included in this document.

The intended audience for this document is the Project Managers and Technical Experts.

Document Control

Date/ Version	Description/ Author/ Contributors/ Distribution
2021.09.18/1.0	Project Proposal / Authors Zakiyyah Noorally, Bernadette Veninata

Terms used in this document

Also see for additional information.

Term	Description
Agile	An iterative software development methodology.
Scrum	Framework allowing team collaboration during software development.
UR QnA	Name of the chat system being implemented.

1. Project Title

UR QnA is a convenient, fast and efficient way to answer student's questions about the University of Regina.

2. Project Description

UR QnA is a web-based chat system that will provide a convenient and efficient platform for University of Regina students to ask questions they have about the University during the COVID-19 pandemic. The chat system provides University of Regina students with the ability to obtain immediate answers to questions relating to the University using an online chatbot. If the chatbot is unable to assist them, users will have the ability to schedule an appointment with a front-desk staff member to chat on a one-on-one basis.

3. Introduction

Live chat software has become an increasingly valuable tool to engage with others online. Live chat systems provide instantaneous global connectivity and have therefore emerged as a leading form of communication, outpacing both email and phone use. Online live chat systems allow users to seek information from an organization through synchronous dialogue with a human service representative via a chat box within a browser. This allows users to receive assistance in a timely manner as, according to freshworks.com, response time is reported to be under 3 minutes for most businesses across all industries.

The integration of Artificial intelligence makes it possible to chat without human interaction using natural language processing (NLP). Human-like conversations can be simulated with users through a chatbot. Intelligent chat systems provide 24-hour service to users. Users may access the system from the comfort of their own homes just by having access to the Internet. According to ChatBot.com, 74% of users prefer chatbots while looking for answers to simple questions.

4. Problem Statement

4.1 Project Motivation

During the COVID-19 pandemic, the University of Regina transitioned to online learning for most students. Remote learning, however, created a unique set of challenges to pre-existing methods of communication.

- Faculty and staff members of the University of Regina were working from home and, therefore, students were unable to have their questions answered in person.
- The voicemail of several university departments was often full and was not accepting any new messages.
- The last resort was to use email, and this was not necessarily the fastest method of communication, usually taking several business days given the large volume of emails received by the University.

Consequently, certain questions requiring simple answers were being answered (on average) a few days later which caused frustration among students.

With faculty and staff working remotely, intelligent live chat technology (in combination with chatbot AI) can serve to alleviate some of these communication bottlenecks by providing the ability to communicate in real-time without solely relying on in-person communications, business hours, emails, and phone calls.

4.2 Problem Definition

The goal of the UR QnA web application is to remedy the communication problem that is negatively impacting a significant portion of the University of Regina student body. The web-based chat system app is being designed to effectively, efficiently and quickly bridge an existing communication gap between students and the University, and more specifically, between students and front-desk staff.

UR QnA serves to alleviate these communication problems through the following process:

- Students, both domestic and international, can have their general inquiries answered immediately via the chatbot (available 24 hours a day).
- If their questions are not fully addressed, the app provides a tool to immediately book an appointment with front-desk staff during regular working hours (8:30 AM to 04:30 PM).
- During the appointment, the student and the staff member can communicate in real-time via the live chat system. The application will also provide a front-desk staff dashboard and

student homepage, where users can easily view details about upcoming appointments.

5. Solution Overview

5.1 Flow Diagram

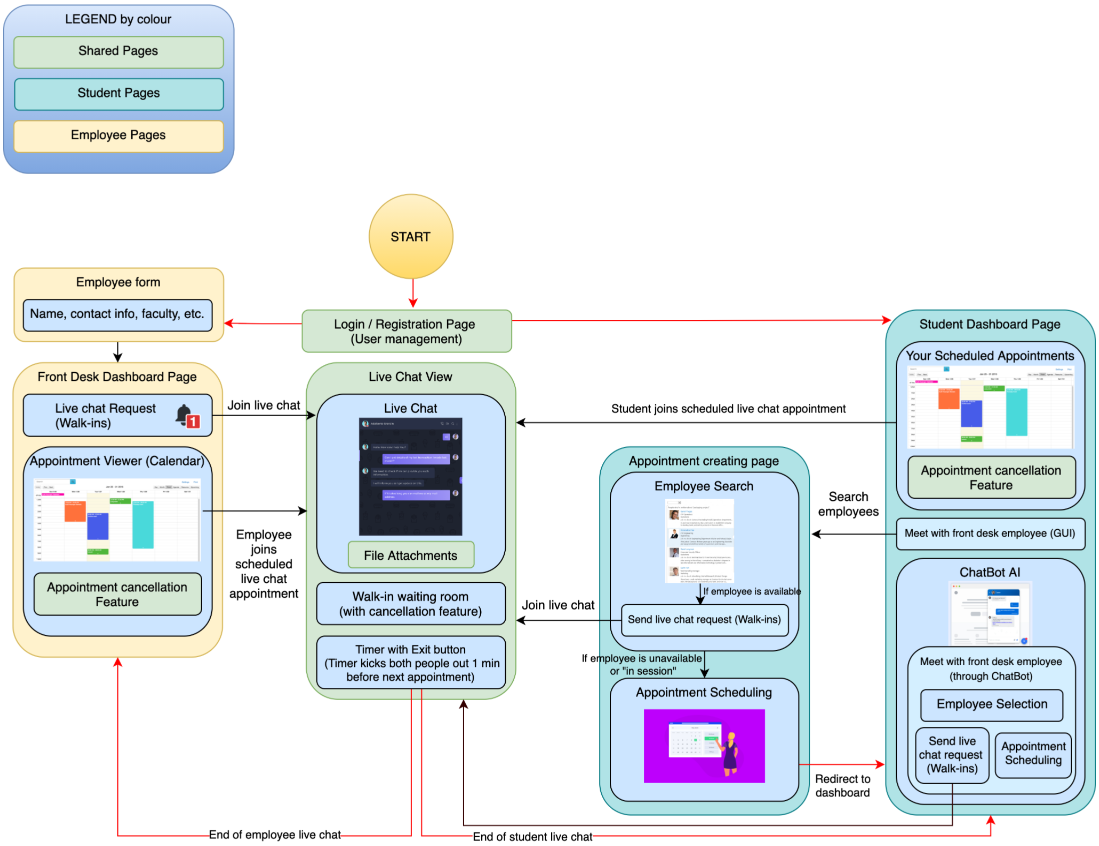

LEGEND by colour
Shared Pages
Student Pages
Employee Pages
START
Employee form
Name, contact info, faculty, etc.
Login / Registration Page (User management)
Student Dashboard Page
Your Scheduled Appointments
Appointment cancellation Feature
Front Desk Dashboard Page
Live chat Request (Walk-ins)
1
Appointment Viewer (Calendar)
Appointment cancellation Feature
Join live chat
Live Chat View
Live Chat
File Attachments
Walk-in waiting room (with cancellation feature)
Timer with Exit button (Timer kicks both people out 1 min before next appointment)
Employee joins scheduled live chat appointment
Student joins scheduled live chat appointment
Appointment creating page
Employee Search
If employee is available
Send live chat request (Walk-ins)
Join live chat
If employee is unavailable or "in session"
Appointment Scheduling
Search employees
Meet with front desk employee (GUI)
ChatBot AI
Meet with front desk employee (through ChatBot)
Employee Selection
Send live chat request (Walk-ins)
Appointment Scheduling
Redirect to dashboard
End of employee live chat
End of student live chat

Figure 1 - Flow diagram

5.2 Solution Overview Details

The UR QnA web application aims to provide assistance to University of Regina students especially during the online learning phase as a result of the COVID-19 pandemic. Our application will allow students the ability to easily connect with front-desk staff remotely using an online chat system, as defined above.

- The application will consist of a login page for both students and front-desk staff members, as well as a registration page which will require a student/employee ID and an @uregina.ca email address for proof of enrollment/employment.
- A chatbot AI will be used to answer simple general student enquiries where intervention by a front-desk staff member is not needed.
- When a student requires an answer to a more specific/complex question, a list of all front-desk staff with their availability
- If the front desk staff is available a student may send a live chat request and wait in the "waiting room" until it is accepted or rejected (with the ability to exit or cancel their request).

- If the front desk staff is not available, students may schedule an appointment with them (via both the GUI or the AI chatbot assistant).
- Front-desk staff members, and students, will both require a personalized view to check their registered appointments and to start a live chat.
- Both students and front-desk staff may cancel appointments.
- If the appointment was cancelled by the front-desk staff, a message will be displayed to the student with cancellation details and a button to delete the appointment from their schedule will show up (and vice-versa for student cancellations). Students can also cancel an appointment should the need for it arise.
- During a live chat session students and staff members will also can send a picture/file attachment.

The application will be designed as a single page application using Vue and utilizing microservice architecture. Each service is separated and will be communicating only via API calls. Each service is hosted by Firebase to manage complexity. Our solution contains components to handle:

- hosting with Firebase Hosting,
- authentication with Firebase Auth,
- data storage with Firebase Firestore,
- file storage with Firebase Storage,
- secure code environment with Firebase Cloud Functions,
- AI chatbot intent handler with Dialogflow by Google.

Each component can talk with each other and is able to work with the same data and set of authenticated users. Each component will have a set of security rules to prevent misuse of resources based on the presence of an authentication token and data about the user stored in Firestore.

6. Functional Requirements

Our app will contain two major user groups/roles:

- Current students at the University of Regina, and
- Front Desk Employees.

Below is a list of functional requirements for this project.

6.1 Current University of Regina student

A current University of Regina student is a student who is already admitted to the University of Regina. Every University of Regina student will be able to use our chat interface by signing up for the service using their official University of Regina (uregina) email address and student ID.

As a current University of Regina student, I want:

1. To register for the chat system (sign-up functionality).
2. To login to my account (login functionality).
3. To use the live chat system (with chatbot AI) to ask questions about the University of Regina (example: When is the deadline to apply for scholarships?).
4. To have my questions answered in a timely manner.
   a. I expect the chatbot to respond to my simple questions.
   b. I expect to talk to a person if the chat bot is unable to answer my questions.
5. To see a list of available staff and their status (“available” or “in session”) so that I know which staff is available to communicate with through the chat system.
6. To chat with an actual person (staff) for complicated questions I may have by
   a. booking an appointment with them
   b. or directly chatting with them if their status is “available” (send a chat request).
7. To be put in the waiting room until front-desk employee accepts my live chat request (with ability to exit or cancel request).
8. I want to be notified when a front-desk employee cancels my chat request.
9. To be able to send a picture/file through the chat system to better explain my questions.
10. To obtain a copy of my interaction with the chat system (chat history).
11. To cancel an appointment with a staff member should the need arise.
12. To see my registered appointments with staff members.

6.2 Front-desk staff members

A front-desk staff member is an employee of the University of Regina. Front-desk members can sign up to use the chat system using the official University of Regina (uregina) email address and Employee ID.

As a front-desk staff member, I want:

1. To register for the chat system (sign-up functionality).
2. To login to my account (login functionality).
3. To answer the questions students have through the chat system by interacting with the students on a one-to-one basis during the regular working hours (8.30 AM to 4.30 PM).
4. To have my status appear as "available" so that students know I am available for a direct chat.
    a. To view live chat requests of users who have not made an appointment.
    b. I want to be able to accept or reject live chat requests made by students.
5. To have my status appear as "in-session" so that students know I am currently busy with another student.
6. To be able to send a picture/file through the chat system if the need arises in order to better assist students.
7. To obtain a copy of my interaction with the chat system (chat history) for my own records.
8. To accept any appointment request being made by a student.
9. To reject any appointment request being made by a student if the need arises.
10. To have a personalized view to check all registered appointments with them, as well as a view to chat with a student

7. Non-functional requirements

7.1. Security, Safety and Privacy Requirements

For the security, safety and privacy requirement, the chat system need:

1. To have login functionality whereby the users are able to access the system only with a valid uregina email, employee/student ID and password.
2. To allow users to change their password
3. To have a 'Forgot Password' feature.
4. To protect the information being communicated between a user and the student

7.2. Usability and Accessibility Requirements

For the usability and accessibility requirements, the chat system need:

1. To have a responsible design, implying users can view the chat system on a desktop/laptop and mobile phone screen.
2. To have a user-friendly interface which is easy to learn and teach.

7.3 Transition requirements

For the transition requirements, the chat system need:

1. To have training videos to teach students how to effectively use the system to obtain answers to their questions.
2. To have training videos/ documentation for staff members to learn how to use the system to assist students.

8. Project Team and Roles

8.1 Project Team Members and Roles

Team Members	Roles and Project Responsibilities
Bernadette Veninata (veninatb@uregina.ca)	**Project Manager** Bernadette is a Project Manager and will collaborate with Zakiyyah at all stages of this project so that she can learn from her co-manager. **Skills:** ● Proficient web development experience in following languages related to project: HTML/CSS, JavaScript, Node.js. ● Good at documenting business requirements and procedures and completing high-level diagrams. ● Designing and delivering presentations. **Responsibilities:** ● Will further study and integrate Firebase, Vue.js, PostCSS and chatbot AI with corresponding Dialogflow component. ● Will use Jira extensively to plan and track the project. ● Will also help Zakiyyah in other aspects of the project deliverables such as the report and presentation.
Zakiyyah Noorally (znz371@uregina.ca)	**Project Manager** Zakiyyah is a Project Manager and will collaborate with Bernadette at all stages of this project so that she can learn from her co-manager. **Skills:** ● Intermediate web development experience (HTML, CSS, JavaScript). ● Good at documenting business requirements and procedures. ● Experience with project management and managing teams for code implementation. ● Designing graphics for visual communication and delivering poster presentations. **Responsibilities:** ● Will work on project deliverables such as the project report and presentation. ● Will help Bernadette with the software development. ● Will also help in the other aspects of the software implementation phase to further broaden her knowledge and skills set. ● Will use Jira extensively to plan the project, delegate work and manage the team.

Sirvan Parasteh (sirvanparasteh@uregina.ca)	**Subject matter expert/Technical Resource** ● Sirvan is the course instructor of this software development course. ● He will be setting certain requirements which the Project Managers need to ensure are being met when implementing this project. ● Sirvan also will coach the Project Managers and advise them whenever they have queries about the project.
Urvish Rana (urh800@uregina.ca)	**Subject matter expert/Technical Resource** ● Urvish is the TA of this course. ● He is available via appointment and will provide technical assistance to the Project Managers if they have any questions during the implementation phase.

8.2 Project Reporting Structure

The Project Managers will meet with Sirvan after successful completion of every milestone to ensure they are on the right path. Additional appointments will be when they need help or advice in implementing a particular functionality. Project Managers are developing this chat system under the agile framework using scums.

8.3 Project Team Status Reporting

The Project Managers will meet weekly at least once (preferably twice) to discuss the progress of the projects and the next steps to be taken.

9. Development Environment

9.1 Software tools

Our project will be developed using VS Code with Vetur, ESLINT, BABEL, and VS Live Share extensions. The Zhs Unix shell will allow for shell scripting to connect with Firebase CLI.

The Firebase Local Emulator Suite and Firebase CLI will be utilized. Firebase CLI will allow for hosting our project, accessing Firestore and Firebase rules (authentication), and to deploy Firebase Cloud Functions.

Version control will be tracked using GIT and changes will be saved to GitHub. Jira will be used for issue tracking.

9.2 Programming Languages

Our application will also be developed in the HTML/CSS, JavaScript, Node.js and Vue.js environments in combination with PostCSS and Dialogflow to develop our chatbot AI.

9.3 Project folders

The below project folders will have all supporting documents that provide background and reference material for the project:

- https://github.com/bernadetteveni/CS-476-Project
- https://cs-476-project.atlassian.net/

10. Project Timeline

10.1 Project Milestones

Following are project milestones:

Milestones	Due Date
Group Formation, 2 students per group	September 10th, 2021
Project Title and Proposal Draft with list of requirements, functional and quality	September 18th, 2021
Design of interfaces (use case and interaction diagrams)	November 12th, 2021
Development Phase	November 12th, 2021
Web application ready to be tested with test cases	November 13th, 2021
Testing Phase	November 21st, 2021
Completion of project report	November 21st, 2021
Completion of student and employee training materials	November 21st, 2021
Completion of presentation	November 21st, 2021
Submission of projects codes, presentation and report	November 22nd, 2021
Project Presentation	November 23rd, 25th, 30th, and December 02, 2021 (depends on the deadline set by the course instructor).
PROJECT CLOSURE	December 02, 2021

10.2 Gantt Chart

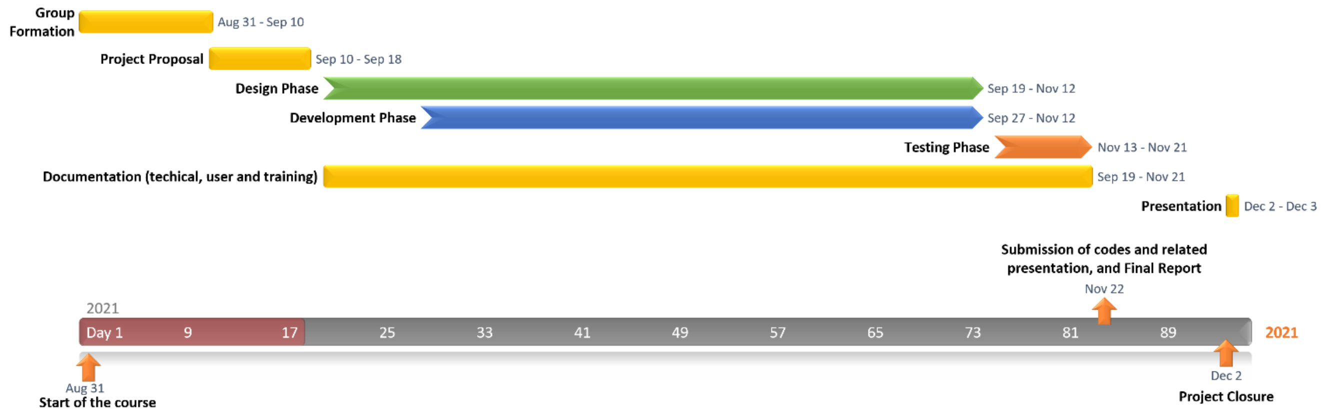


Figure 2 Gantt Chart

11. Expected Outcomes and Products

11.1. Deliverables

Deliverables	Description
Project Proposal	A document including problem statement, project introduction, description, business requirements, timeline, expected product outcomes.
Diagrams	Use case diagrams and user interaction documents before the implementation phase.
Code	All codes (including front-end and back-end) which has been written to implement the program
Working deployed version of UR QnA	The initial deployed version of UR QnA with basic functionality implemented.
Testing document	A document including test cases and the result of testing.
Procedure Documentation	Documents including details about the implementation of the code, business flows and procedures.
Training Documentation	A document showing the users how to use the chat system
Final Report	A document including detailed analysis of work completed over the course of the semester, future work, conclusion.
Poster Presentation	A presentation summarizing the work accomplished during the term. This presentation will be presented in front of the class.

11.2 Risks and Constraints

11.2.1. Constraints

Project completion date is no longer than the deadline set to submit the codes and the presentation (currently November 22nd, 2021).

11.2.2 Initial Risks

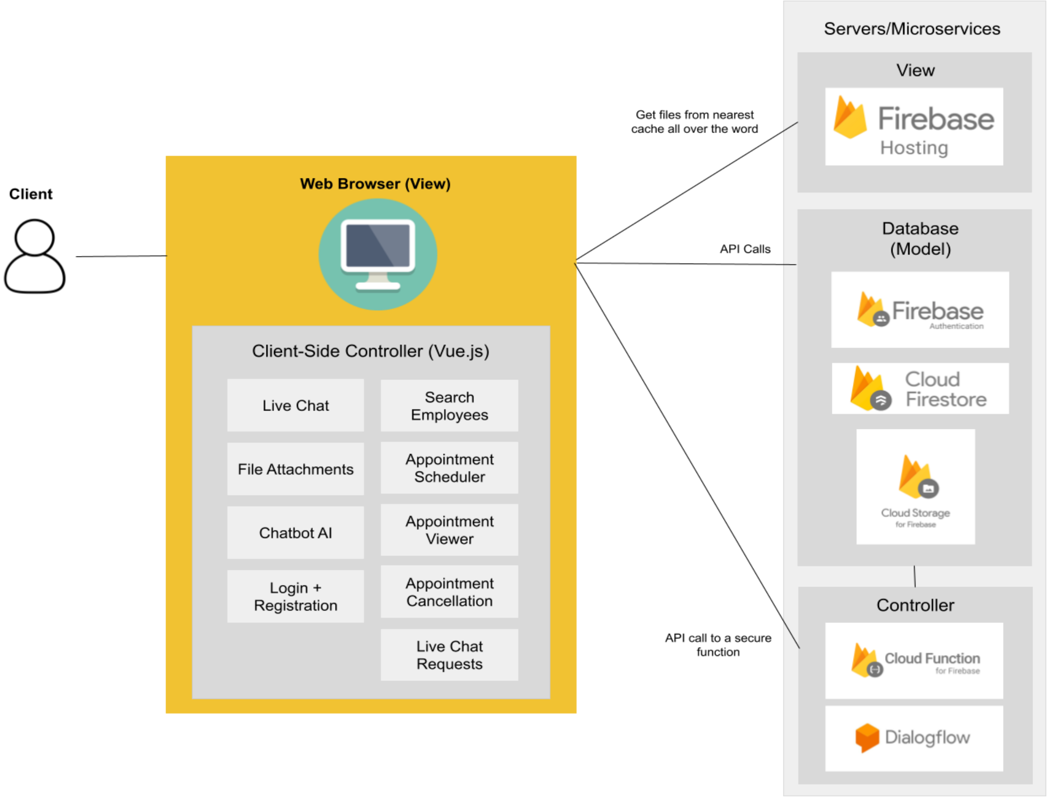
Given the time constraints of only two months to design, implement and present this project, an initial version of UR QnA will be implemented. UR QnA 1.0 will contain at least the basic functionality of a personalized web portal with a chat system. Other functionalities will be added if time permits.

However, this may be at the expense of certain non-functional requirements.

- Additional verification needs to be carried out by an Admin (additional stakeholder) of the system to verify before the employee role of everyone who sign-ups as an advisor before granting a valid username and password to use the system as front-desk member.
- To maintain respect for everyone, it is important to make sure any student does not use inappropriate words. There should be a mechanism in place to report such behavior.

For simplicity's sake, we have ignored these additional requirements in the initial implementation of UR QnA. These will be further discussed in the future work of the final report.

12. High Level Diagram


Servers/Microservices
View
Firebase Hosting
Get files from nearest cache all over the word
Client
Web Browser (View)
API Calls
Database (Model)
Firebase Authentication
Cloud Firestore
Cloud Storage for Firebase
Client-Side Controller (Vue.js)
Live Chat
Search Employees
File Attachments
Appointment Scheduler
Chatbot AI
Appointment Viewer
Login + Registration
Appointment Cancellation
Live Chat Requests
Controller
API call to a secure function
Cloud Function for Firebase
Dialogflow