SE 3XA3: Software Requirements Specification
QChat - QuestionChat

Team #14, QChat
Adit Patel - patela14
Harsh Patel - patelh11
Vrushesh Patel - patelv12

October 6, 2017

Contents

List of Tables

Table 1: **Revision History**

Date	Version	Notes
06/10/2017	1.0	Inital Version
20/11/2017	2.0	Updated mandated constraints and Terminology
22/11/2017	3.0	Fixed and updated Individual Product Use Case and Functional Requirements
25/11/2017	4.0	Updated The Scope of Work and the Product and the Context of Work

This document describes the requirements for QChat. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**). Our modification to the template includes combining the subsubsections of 1.2 to form one subsection that combines the client, customers, and stakeholders.

1 Project Drivers

1.1 The Purpose of the Project

The purpose of our project is to allow students who are nervous to ask questions in large lecture rooms to have a voice through an anonymous platform by providing a cross platform Q and A app. The app will allow users to anonymously post questions. Users will be able to upvote questions they would like to find the answers to and the most popular questions will be showed at the top for the professor to answer.

1.2 The Stakeholders

The clients for this project are Group 14: QChat and Asghar Bokhari. Professor Bokhari is the client who asked for a project to be done with broad requirements. Group 14: QChat is the client who decided the specific project and provided specifications for the end result. The customers are the students and the professors at McMaster University. Students and Professors will be our customers because this application will directly impact these two entities as it is a tool of communication between them. They will making the choice of using the application on their devices. As our problem is an issue that occurs in the university ecosystem, our stakeholders will be members of this ecosystem. Our stakeholders are students, professors, developers of this project, researchers interested in the outcome of this project. Students will be impacted heavily as they will move to a friendly platform to ask questions to colleagues and/or the professor. As students use this platform to interact in a new way, professors will be impacted heavily as well. Being expected to resolve questions through/from this platform may need them to reform the way they deliver lectures to allocate enough time to review and answer questions from the platform. The developers of this project are required to maintain the application and provide it new features and make it extendable. The developers will need to resolve issues that the application has and may face learning curves to do so. Researchers are also a stakeholder as the application can be potentially used to change the way lectures are delivered. A breakout in a new methodology of learning will result in validation of this method being better by means of collecting data or interviews.

1.3 Mandated Constraints

Description: The application should run on web browsers that allow JavaScript to run through a website hosting server.

Rationale: The client will not be using outdated version of web browser when trying to access our application.
Fit Criterion: Our app will run on web browser that have JavaScript enabled, if the user has it disabled, it will ask them to turn it on.

Description: The app will only run when connected to internet
Rationale: The user should be connected to the internet all the time in order to use our app, so that posting and fetching of questions and answer from the database can work
Fit Criterion: the app load automatically if the user is connected to the internet, if not any web browser will give error.

Description: The app should resize on its own to fit on all devices with different screen sizes and orientation, no elements of the UI should over lap with other elements.
Rationale: The user can resize their browser window and the app will resize itself until it hits its boundary where it cannot get any smaller or bigger
Fit Criterion: The maximum and minimum size will be determined based on the size required by the app.

The following are some other constraints have been identified for QChat:

- A web-hosting server with a domain name would a partner or collaborative applications if the app gets released. For the app to run some of the Off-the-Shelf software that it will use are, a web browser, JavaScript and a device like phone, computer etc. where the web browser can run.

- The anticipated workspace environment for the app is anywhere, where the user has access to internet and a device that has web browser.

- The product will be finished and ready to be released by December 6th, 2017.

- The operating budget for our project is $0 because all the resources need to build this project are available for free, therefore no funding is needed.

- The application should be available for free to all users.

1.4   Naming Conventions and Terminology

It is very important to make sure that the code in any code base is clear and conforms to a standard set of rules. This allows for ease in understanding of how the code actually works and communication between different coders. Naming and formatting will be done using regular conventions we learned in 2XA3, all our variables will be named using camel notation, classes will be named with capital.

Terminology

App/Application/QChat - the product that is being described here; the software system specified in this document
Project - tasks that will lead to the production of QChat or the application itself.
Client - the person or group of people for which QChat application is being built.
User - group of people who will be using QChat.
Developers - the team building QChat.
Web Technologies - the method by which computers communicate with each other.
JavaScript - part of web technology, programming language that runs on browser.
Firebase - Firebase is googles free real time database and backend service. We will be using this as our database.
Stakeholders - any person who is going to be using the system and is not a developer.

1.5   Relevant Facts and Assumptions

Assumptions

- Assuming all the users have experience using at least one successful messaging platform like Facebook, Instagram, Twitter, IMessage etc.

- Users will be respectful to each other while using the app and will not use the anonymous feature to bully or troll the chat room. Such posts will be deleted.

2   Functional Requirements

2.1   The Scope of the Work and the Product

The scope of our software project is to provide cross platform Q and A app. The app will allow users to anonymously post questions. Users will be able to upvote questions they would like to find the answers to and the most popular questions will be showed at the top, user will also be able to downvote a question. To accomplish this we will use a database and a client/server model. Users will also be able to anonymously answer questions. The technology stack we will be using is Googles Firebase database for the backend and an app to serve the client end. The application should be available for free to all users. Furthermore, the application will need both a device that has a web browser and access to Internet.

2.1.1   The Context of the Work

1. Programming Environment - All developer PCs will need an IDE to write and edit the code, and a web browser to test the application.

2. Connection to the internet - All stakeholders associated with the application will need access to the internet (provided by an ISP) to access the application in its development and finished product state.

3. Firebase Backend - Needs to be up and running in order for application to work in its development and finished product state.

2.1.2   Work Partitioning

Event Name	Input	Output
Create a database	Googles Firebase Database	Googles Firebase
Add data to database	Developer Code	Googles Firebase
Fetch data from database	Developer Code	Terminal
Add functionality to upvote	Developer Code	Terminal
Create UI	Developer Code	Web Browser
Connect UI with Adding and Fetching data from database	Developer Code	Web Browser

Table 2: **Work Partition Table**

2.1.3   Individual Product Use Cases

Product use case: Post anonymous question in chat room
Actors: Student
Input: Type question as text
Output: Question appears in chat room anonymously

Product use case: Answer question
Actors: Student/Teacher
Input: Click on question, add reply, submit
Output: Answer appears next to question in chat room

Product use case: See top questions in chat room at the current time
Actors: Teacher, Student
Input: Click on top questions button
Output: New page appears with top 5 questions students are currently upvoting

2.2   Functional Requirements

Description: App should allow users to post questions and answers to chatrooms anonymously
Rationale: This is our primary requirement that also serves as the main functionality of our web app

Fit Criterion: Test if QChat is able to push to database and later pull it back onto the UI

Description: App should be able to run on all web browsers and devices that have access to Internet
Rationale: We want our app to the accessible across all platforms
Fit Criterion: Check if app runs on Chrome, Mozilla Firefox, Internet Explorer, iOS, Android

Description: User should be able to connect to different chat rooms by giving a unique ID
Rationale: Each room will have a unique ID so that user can join and the data (all questions and answers asked by users) associated with that chat room will remain in that chat room
Fit Criterion: Check if data is associated with specific unique ID and can be accessed only when given that specific ID

Description: User should be able to upvote and downvote questions which will push the questions and answers with most upvotes at the top of the chat
Rationale: This will allow professors to easily answer the question that is common among students
Fit Criterion: Upvote different questions and see if top questions post changes dynamically

3 Non-functional Requirements

3.1 Look and Feel Requirements

Since the final product is going to be a web application, the product should look visually appealing, so that our users find the product usable. It should look professional to gain users trust to make them feel like it's reliable.

3.2 Usability and Humanity Requirements

Our web app should have an easy to use interface which is simple and intuitive. It will be easy for our users to understand how to use our app because we will have similar UI and UX as other successful messaging platforms. The app will also provide an ease of access as it will run on an internet browser and will not require the user to download anything. The app will be tested before it is released therefore there will be very minimal to no errors allowing user to have a smooth experience using our app. Meaning one months use of the product should result in a error rate of less than 1 percent. Our team plan on conducting a quick survey after a month after release to find about the user experience from real users.

3.3 Performance Requirements

QChat is a realtime messaging app because of this a key functionality should be the fact that the app should take less than 3 seconds to load on a web browser. Another performance requirement is there should not be a long lag between posting something to the app and it showing up/updating on all other places. The app will be available for use 24 hours per day, 365 days per year. If the app crashes this should not delete the data, data will be safe. The app should not have any glitches when running on different web browsers. For example any UI glitches.

3.4 Operational and Environmental Requirements

This section describes the physical environment in which our app will run. QChat will operate on any web browser and will need access to the internet. The app might get new features in future but the update should not affect the core functionality of the app and will not cause any previous features to fail. The app should work on all browsers without any glitches and should work on all devices that has access to internet and web browser. The product will be distributed as a website link to all users.

3.5 Maintainability and Support Requirements

Our team will use several techniques to quantify the time necessary to make specified changes to the product. This will include using Gantt chart to create goals and deadlines and using scrum methodology to make sure were hitting those targets. After completion of development, the team will test the application on a daily basis and make modifications as required or requested. Users will provide feedback sooner or later and our team will support them by bringing the necessary changes to provide a better experience.

3.6 Security Requirements

Any software project need security requirement to ensure the safety and integrity of the product. As such here are a few security requirements that we will integrate, only the developers will be allowed to edit the backend and frontend of the application.Only the developers will be allowed to add or take away features from the application. Only the developers can see where and how the data will be stored, the application should protect itself and the data from hacking or any intentional attacks. This will be done with gitlab authentication, since our repository is on gitlab only members with access to that repository will be able to add to and alter the code there. Second, since the purpose of this application itself is to allow anonymous Q and A, privacy of the users will be kept safe, in no way will the app be able to trace back who posted what comment and message. If any privacy policy changes occur in future, all the users will be notified about it.

3.7   Cultural Requirements

In this section we describe any sociological factors that may affect the acceptability of our product. QChat will use the English language since most to all of our initial target users in McMaster university speaks english. The app will not use any offensive or vulgar language.

3.8   Legal Requirements

There are no legal requirements that the team is concerned with at the moment.

3.9   Health and Safety Requirements

Our QChat app is a web service, there are no health and safety issues when users use the application as expected from an ideal user.

4   Project Issues

4.1   Open Issues

Here we list a couple of open issues that have been raised by the team that have not yet come to a conclusion. One major open issue is whether to keep the data created by the users permanently or whether we should delete it after 24 hour like Snapchat. This can affect security requirements and functional requirements. This issue also has a major cost-trade off we would need to consider. If we store the data we would eventually need to pay for server space, if we do not then we don't necessarily have to pay but it would mean some features such as search in past class would not exist. Another open issue is whether to provide the service in the form of a mobile application or not, this topic has not been dealt because we have limited time for the development on our application and are currently focused on the web app portion only.

4.2   Off-the-Shelf Solutions

QChat will be built with Google Firebase, this is a free off-the-shelf realtime database service that Google provides for developers. QChat will also use the angular framework to help us quickly setup and launch out web app.

4.3   New Problems

There are no new problems that the team has not addressed.

4.4 Tasks

Task	Role	Timeline
Model Implementation	Software Engineers	Mid October
Backend Implementation	Software Engineers	Early November
UI Implementation	Software Engineers	Mid November
Test and Review	Software Engineers	End of November

Table 3: **Tasks Table**

Model Implementation will include making a model for storing the user data (question and answers). This phase is important because no relevant details can be left out. It is the backbone of our project because our whole app is based on adding and fetching the data from the database. Backend + features and UI implementation will include setting up all our backend push and pull methods along with the front end display methods to create the core functionality of our project. Test and Review to ensure our app is working without any bugs and glitches.

4.5 Migration to the New Product

The development team is creating version 0 of the QChat app, therefore there will be no need to migrate to a new product.

4.6 Risks

- Server failing to request the data from our database if too much traffic comes on our web app, probability of it failing is really low because the app is only for users (students and professors) at McMaster University, since the servers can handle several thousands of requests this shouldnt create a problem.

- Users using the anonymous platform to send disrespectful messages and trolling the chatroom, a high probability of this occurring. Would need to address the issue before shipping.

4.7 Costs

The development cost will be the number of hours spent developing the web app. Our team estimates every week each member will spend approximately 8 hours working on the project. This amounts to about 24 hours/week per team working on this app and the documents, for 12 weeks. Our monetary budget for the project is $0.

4.8   User Documentation and Training

A software is only as useful as the documentation and training it provides. QChat is a front facing web application for the user and not a business to business to service so we do not need something like an API for other developers and user documentation.

4.9   Waiting Room

None as of now.

4.10   Ideas for Solutions

None as of now.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.