Software Requirements Specifications

THOTH

1.0.0

**Prepared For:**

*Hashedin.com*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 1.0 | 03-03-2020 | Ashwini Abhishek | Initial release |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Anphy Joseph | Agile Coach + Tech mentor | 03-03-2020 |
|  | Aniket |  |  |

**Table of Contents**

1. Introduction

1.1. Purpose

1.2. Scope

1.3. References

1.4. Targeted Users

2. Overview

3. Objective

4. Functional Requirements

4.1. Error handling

4.2. User Characteristics

4.3. Feature Priority

5. System Architecture

6. Assumptions and Dependencies

7. Constraints

8. Known Issues

9. Use cases

10. Non functional requirements

10.1. Performance

10.2. Availability

10.3. Security

10.4. Maintainability

10.5. Browser support

10.6. Devices supported

10.7. OS Support

11. Testing

12. Deployment

13. Out of Scope

14. Change Management Process

15. Appendices

15.1. Appendix 1

15.2. Appendix 2

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 1. Introduction

*A plug and play chat module specifically aimed at handling high volumes of transaction along with the capability of push notifications / in-app notifications.*

## 1.1. Purpose

To help any website or an app to have a functionality of chat by using our API i.e to build an embeddable chat module

## 1.2. Scope

*Users would be able to*

1. *Participate in an active chat session between some specific users involved in that activity.*
2. *Can see their chat history after the session is over and decide whether to show chat history or not.*
3. *Have a dashboard with profile photo and customizable UI.*

## 1.3. References

## 1.4. Targeted Users

1. *Companies who need a chatting application embedded to their webpage or an app, so that their user can easily communicate with each other for any purposeful interaction.*

# 2. Overview

*An interactive api to help the users to chat between each other easily for any purposeful activity .*

*Active chat sessions are made according to the needs of any user which he/she is involved in, like if you booked an uber for yourself then a chat session is created between you and uber driver.*

# 3. Objective

*A plug and play, chat module specifically aimed at handling high volumes of transaction along with the capability of push notifications / in-app notifications. The notification payloads should be customisable*

# 4. Functional Requirements

* *Ability to have 1:1 conversation*
* *Ability to have 1:n conversation*
* *Ability to send/share attachments - like file/photo/location*
* *Customizable UI*
* *Ability to send push notifications*
* *Ability to customize notifications as per various needs*
* *UI should partition chat based time*
* *Message sent/delivered time along with update in UI*
* *Ability to enable/disable chat history. Basically a config flag which determines whether to show previous chat history or not*
* *UI should have profile pics*
* *UI can have a switch type component to display all the users/rooms which makes switching between different users bit modern and easy*
* *Dashboard - Ability to fetch chat history*
* *Dashboard - Ability to run analytical queries over all the chat histories. It could mean - How many times a particular word appeared.*
* *Dashboard - Ability to blacklist a user*
* *UI - report another user or block the user*
* *Dashboard - Ability to archive chat history*

## 4.1. User Characteristics

* *Message in an active chat session and share attachments like file/photo.*
* *Customise UI according to the choice for example dark-mode*
* *View previous chat history and is also able to disable chat history.*
* *UI should have profile pics.*

## 4.2. Feature Priority

1. Ability to have 1:1 conversation and 1:n conversation
2. Ability to send/share attachments - like file/photo/location.
3. Message sent/delivered time along with update in UI.
4. UI should partition chat based on time
5. Ability to send push notifications and *customize notifications as per various needs.*
6. Dashboard - Ability to fetch chat history
7. Dashboard - Ability to blacklist a user
8. UI should have profile pics
9. UI can have a switch type component to display all the users/rooms which makes switching between different users bit modern and easy
10. Dashboard - Ability to run analytical queries over all the chat histories. It could mean - How many times a particular word appeared.
11. Enable bot support. Should be configurable
12. Auto replies from bot based on words. [This would need ML. It should be picked up only

if all other features have been developed]

1. Ability to mask/hide abuses/cuss words.

# 5. System Architecture

*Provide a logical diagram of the proposed system. In case there are certain parts that do not fall under HashedIn’s scope, highlight them accordingly.*

# 6. Assumptions and Dependencies

* *Depends on the data of the users provided by the host system i.e system which is using our api. For example if uber uses our chat api then the data about the active users,current driver for a particular session will be provided by the uber over which our api will add the chat functionality*

# 7. Constraints

* The system need to be permanent connected with internet
* Chatting in a previously closed session is not possible

# 8. Known Issues

*Highlight issues arising from system design that the client should be aware of and is discussed and agreed upon. Eg: Because of a certain given constraint*, ...

# 9. Use cases

1. *Can be used in various apps for chat functionality.It would be like a plug n play chat functionality provided to the various apps.One use case would be to use it for chatting in customer support section*
2. *Can be used for chatting between multiple users who are sharing a ride using carpooling apps like QuickRide*
3. *Can be used by Uber for chatting between Driver and Passenger.*

# 10. Non functional requirements

The following are the non-functional requirements that are important for the software application.

## 10.1. Performance

*Application must be lightweight and must send message instantly.*

## 10.2. Availability

*Live chat sessions will be available to active users for that particular activity only.*

## 10.3. Security

*Are there any security consideration that need to be fulfilled. If it is a mobile app does it require anything specific to be considered before publishing the app. If there are payment gateways, consider masking of content - card numbers, If it is a web app , browser security*

## 10.4. Maintainability

*Is there a maintenance period that has been signed off? Are we expected to fix the production issue? What is the understanding of how the deliverable will be managed after UAT?*

## 10.5. Browser support

*Supported browsers : Internet Explorer 11 , Google Chrome, Mozilla FireFox and Safari.*

## 10.6. OS Support

Windows, Linux, MAC OS, Android above 4.4(kitkat) and IOS.

# 11. Testing

*A brief testing plan and high-level timelines. Mention it explicitly if it does not fall under our scope.*

# 12. Deployment

*Details about where and who will provide the environments for dev, testing and production. This could contain details about setting up CI as well.*

# 13. Out of Scope

Mention every little details that is not in scope.

*Payment Gateway*

*Order History*

*Scheduling of Appointments/Orders*

# 14. Change Management Process

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

# 15. Appendices

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## **15.1. Appendix 1**

## **15.2. Appendix 2**