SOFTWARE REQUIREMENTS SPECIFICATION

For Chorus Project

Bilal Özlü 1942614

Ali Şimşek 2099752

Version 1.1 (18.03.2018)

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose of the System

Chorus Project is a Google Hangouts chat bot powered by crowdsourcing. A group of crowd workers will collectively chat with you and try to solve your problems. It aims to enable a conversation with a crowd of workers in order to leverage human computation in a variety of new ways.

1.2. Scope

- The project will contain a collaborative reasoning system which lets workers select reasonable responses from a number of crowd-produced suggestions. This system chooses the best responses to be forwarded to user and removing responses that are not related to the topic of the conversation.
- The project will contain a dynamic scoring system which rewards workers by determining their contributions to the conversation.
- The project will contain a curated memory system which allows memory workers to promote some lines from conversations to a shared memory, which is used to get important facts from past conversations with user.
- The project will contain an interface for memory workers to make it possible for them to edit working memory.

1.3. System Overview

1.3.1. System Perspective

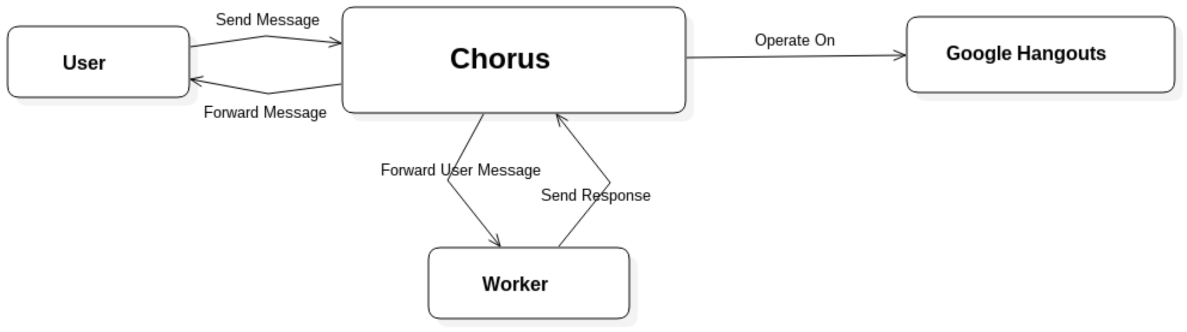


Figure 1.3.1.1: Context Diagram for Chorus System

Chorus is a crowd-powered conversational assistant that allows a user to receive assistance on any online task through a two-way natural language conversation. Chorus works on Google Hangouts to communicate with user. Chorus is capable of performing any task that may be accomplished over the web using information that the user is comfortable sharing.

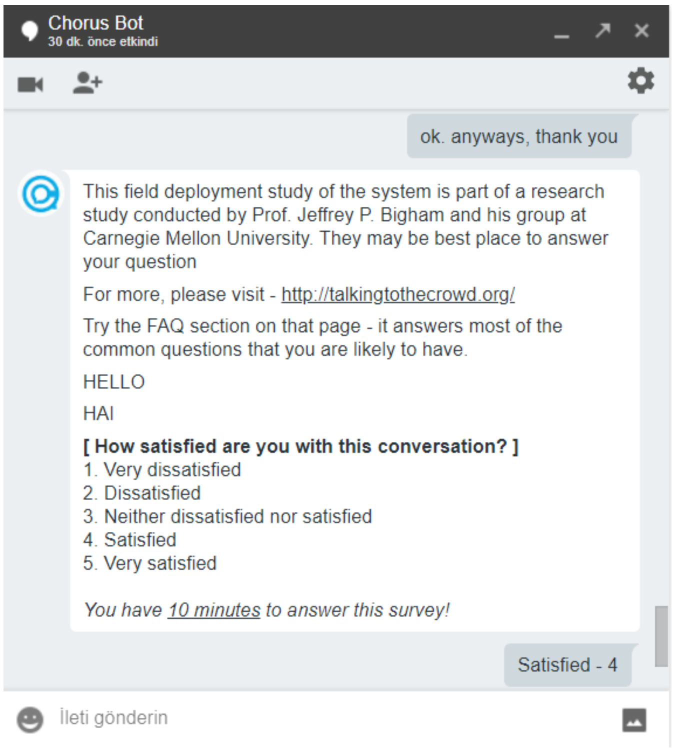
1.3.1.1. System Interfaces

- Chorus system is reachable via Google Hangouts only. Thus, it needs to communicate with Google Hangouts to be able to establish communication between users and workers.
- There's an authorization interface. When a user wants to connect to Chorus via Google Hangouts, he/she needs to fill a form which requires some personal information. This process occurs only once. After the first time, authorization process is done by the Google automatically.

1.3.1.2. User Interfaces

Chorus uses Google Hangouts' user interface since Chorus Bot looks like another user in your Hangouts contacts list. Chat's user interface consists of a button to send photos. User can also draw something via this button and send it. Other than that, there's a section next to that button where you type your text message.

Here is a screenshot of Chorus chat window on a web browser (Figure 1.3.1.2.1).



ok. anyways, thank you

This field deployment study of the system is part of a research study conducted by Prof. Jeffrey P. Bigham and his group at Carnegie Mellon University. They may be best place to answer your question

For more, please visit - http://talkingtothecrowd.org/

Try the FAQ section on that page - it answers most of the common questions that you are likely to have.

HELLO

HAI

[How satisfied are you with this conversation?]
1. Very dissatisfied
2. Dissatisfied
3. Neither dissatisfied nor satisfied
4. Satisfied
5. Very satisfied

You have 10 minutes to answer this survey!

Satisfied - 4

Figure 1.3.1.2.1: Chorus chat window

Here are screenshots of Chorus on mobile devices. From the Figure 1.3.1.2.2, we can see that Chorus Bot is just another contact in Hangouts' contact list.

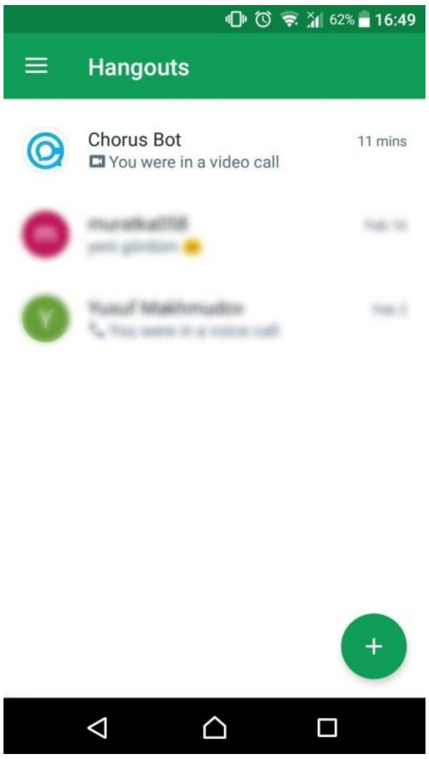
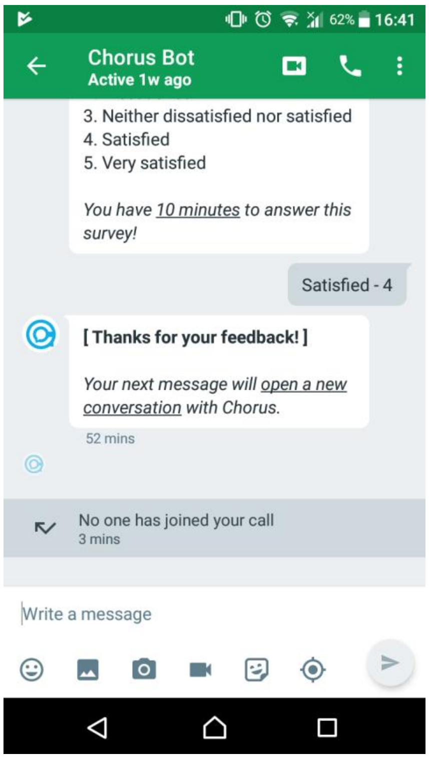


Figure 1.3.1.2.2: Mobile1



3. Neither dissatisfied nor satisfied
4. Satisfied
5. Very satisfied

You have 10 minutes to answer this survey!

Satisfied - 4

[Thanks for your feedback!]

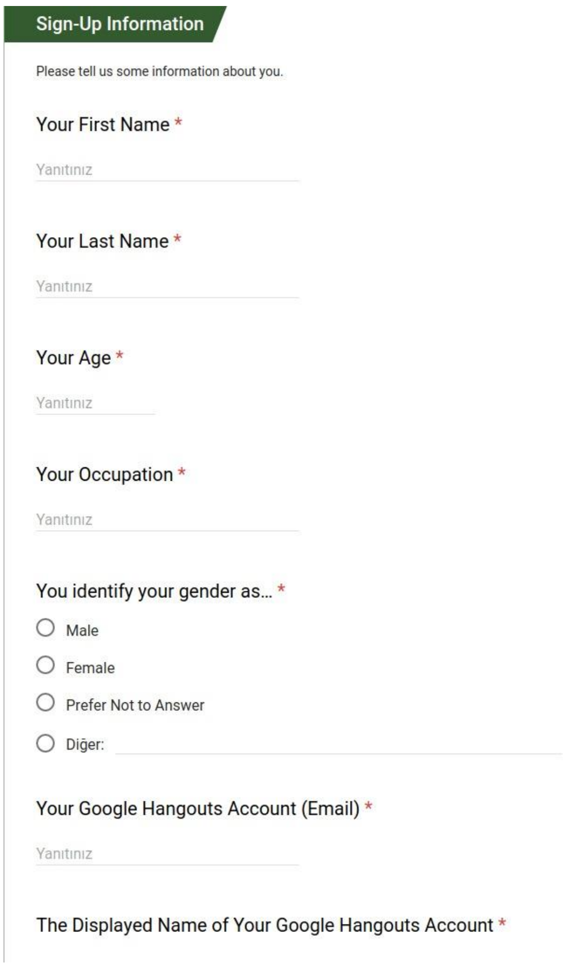
Your next message will open a new conversation with Chorus.

52 mins

No one has joined your call
3 mins

Figure 1.3.1.2.3: Mobile2

The figure at the right side is the registration interface of Chorus (It can be reachable from the Chorus' website : talkingtothecrowd.org)



Sign-Up Information

Please tell us some information about you.

Your First Name *

Yanıtınız

Your Last Name *

Yanıtınız

Your Age *

Yanıtınız

Your Occupation *

Yanıtınız

You identify your gender as... *

- Male
- Female
- Prefer Not to Answer
- Diğer:

Your Google Hangouts Account (Email) *

Yanıtınız

The Displayed Name of Your Google Hangouts Account *

Figure 1.3.1.2.4: Registration Window

1.3.1.3. Software Interfaces

Since Chorus works on Google Hangouts, we won’t use any external software other than what Google Hangouts uses. We will only use a DBMS to create a shared memory section to store some facts from the past conversations with user.

1.3.1.4. Communications Interfaces

In mobile devices, TCP protocol will be used for reliability and security reasons. For the web browsers, HTTP/2 protocol will be used since its latest version and is supported by major web servers and browsers.

1.3.1.5. Memory Constraints

We do not need to make a big issue out of it as we assume system engineering is already handled for the Chorus system as a whole.

1.3.1.6. Operations

The operations are supported by the Chorus system are as follows:

User operations

- Register
- Give feedback
- Send message
- Send photos

Worker operations

- Submit response
- Edit working memory
- Vote for suggested responses
- Vote for important facts in memory
- Browse working memory
- Browse conversation history

1.3.2. System Functions

1-) Register:

When a user wants to use the Chorus for the first time, he/she has to register to the system. This operation happens only once.

2-) Give Feedback:

User is given a feedback form after the conversation ends. User is expected to submit his/her feedback in 10 minutes. Otherwise, the feedback is discarded.

3-) Send Message:

User can send message.

4-) Send Photos:

User can send photos via a button in the chat menu. He/she also can draw something and send it.

5-) Submit Response:

Workers can submit their responses. However, among submitted responses, the top rated response will be forwarded to user.

6-) Edit Working Memory:

Memory workers can edit working memory by adding some lines from conversation with the user or by adding their own summaries of facts.

7-) Vote for suggested responses:

Each worker vote for the suggested answers. Among them, only top rated answer is forwarded to user.

8-) Vote for important facts in the memory:

Memory workers can vote for facts in the memory they think are important.

9-) Browse working memory:

Workers can browse working memory to get some facts about the user and submit their responses according to those facts.

10-) Browse conversation history:

Memory workers can browse conversation history and make additions to working memory from past conversations.

1.3.3. User Characteristics

Chorus project will be used by end users and crowd workers. We can categorize crowd workers as chat workers and memory workers.

End users are required to have Google accounts to be able to register to Chorus system. Then they can have a conversation with Chorus Bot anytime they want. Users can receive assistance on any online task through a two-way natural language conversation. Users need to be an English speakers since all the conversations will be in English.

Workers are not required to have many technical skills. They're only expected to have some education in the use of the interface. To ensure that, workers are expected to complete two guided tutorials before they can participate in the main task. Memory and chat workers go through different tutorials since their tasks are different.

1.3.4. Limitations

Regularity policies: Any information related to user and workers are hidden. None of the users has to reveal their personal information unless they want to.

Hardware limitations: Since Chorus can run any device that runs Google Hangouts, hardware limitations is not a big issue.

Interface to other applications: Chorus' interactions with other applications are limited by the Google Hangouts. Chorus is not compatible with any other system other than what Hangouts supports.

Parallel operation: Since Chorus is only a chatting application, parallelization is not a big issue.

Audit and Control functions: There won't be any audit and control functions.

Higher-order language requirements: For the browser application css, html and javascript will be used.

Signal handshake protocols: In mobile devices, TCP protocol will be used for reliability and security reasons. For the web browsers, HTTP/2 protocol will be used since its latest version and is supported by major web servers and browsers.

Quality requirements: Every chat worker can submit a response. But among them, only top rated response will be forwarded to user to provide quality service.

Criticality of application: Criticality is not a big issue for the Chorus system since it doesn't contain any critical information.

Safety and security considerations: Security of the accounts are managed by the Google. Also, personal information of all workers and users are hidden for maximum confidentiality.

Physical/mental considerations: There are no limitations for physical and mental considerations. Anyone that can use a computer or a mobile device can use Chorus.

1.4. Definitions

Chat Worker: A worker who contributes to conversation with user by submitting response and voting on suggested responses.

Memory Worker: A worker who manages the working memory by browsing past conversations and adding important facts to the working memory.

End User: Person who uses Chorus for personal assistance

DBMS : Data Base Management System

HTML : Hypertext Markup Language

CSS : Cascading Style Sheets

HTTP: Hypertext Transfer Protocol

TCP: Transfer Control Protocol

HTTPS: Hypertext Transfer Protocol Secure

TLS: Transport Layer Security

Encryption: Process of encoding a message or information in such a way that only authorized parties can access it

Mturk: Amazon Mechanical Turk

2. References

- International Standard ISO/IEC/IEEE 29148 Systems and software engineering - Life cycle processes - Requirements engineering https://standards.ieee.org/findstds/standard/29148-2011.html
- Chorus: A Crowd-Powered Conversational Assistant by Lasecki https://dl.acm.org/citation.cfm?id=2502057
- “Is there anything else I can help you with?”: Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent https://www.cs.cmu.edu/~tinghaoh/pdf/2016/2016_hcomp.pdf
- Website of Chorus https://talkingtothecrowd.org/
- https://en.wikipedia.org/wiki/Encryption

3. Specific Requirements

3.1. External Interfaces

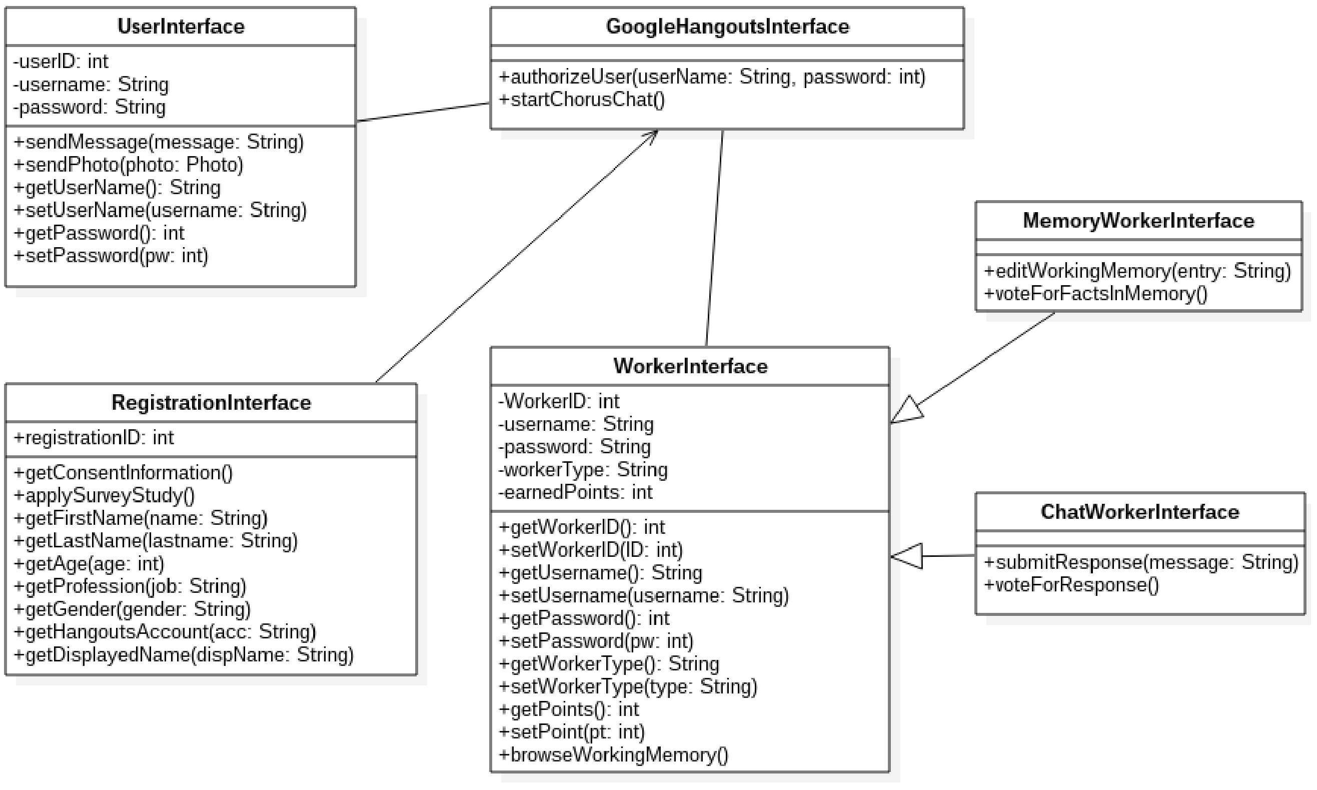


Figure 3.1.1: Class Diagram of Chorus

3.2. Functions

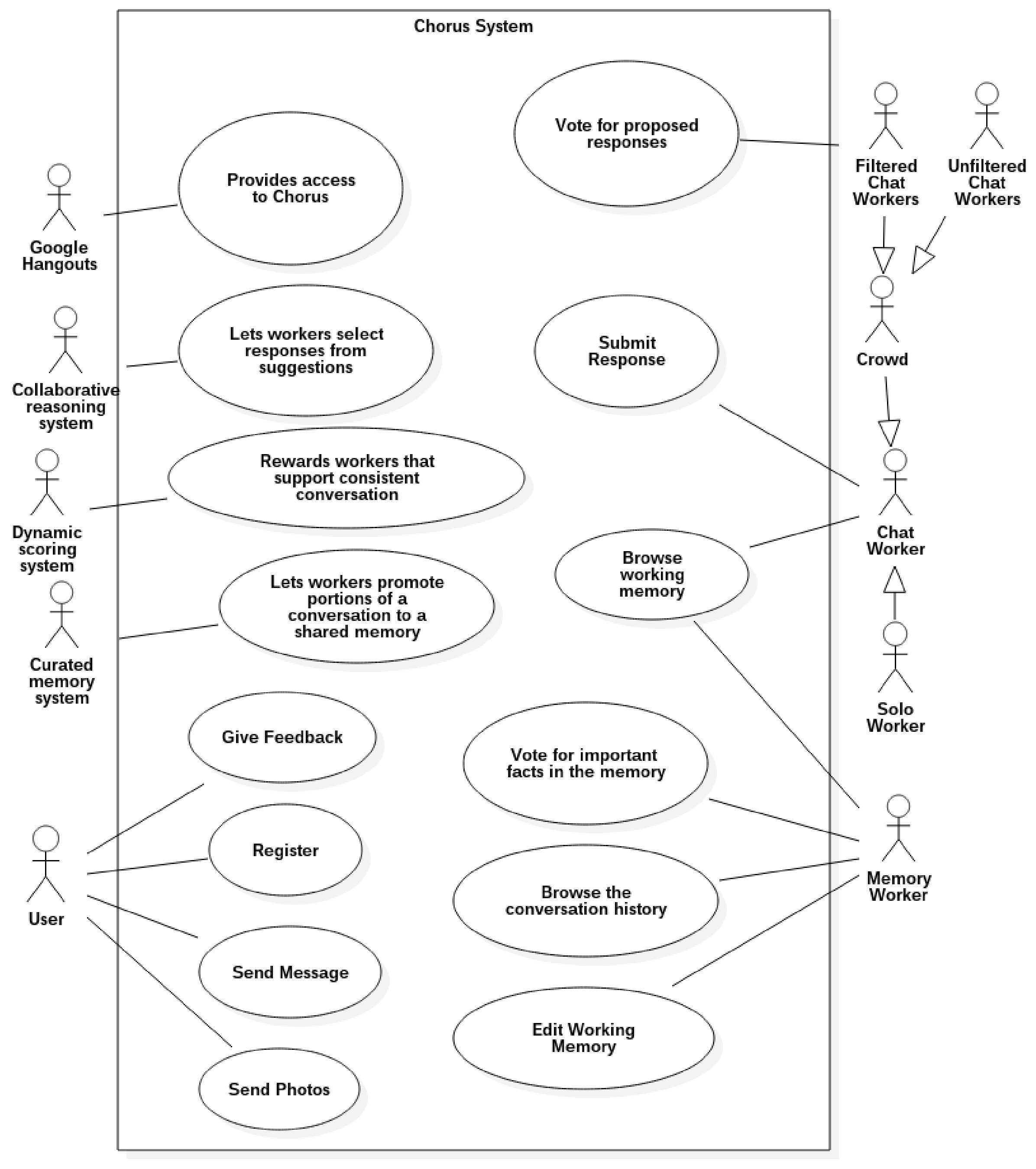

Chorus System
Vote for proposed responses
Filtered Chat Workers
Unfiltered Chat Workers
Provides access to Chorus
Google Hangouts
Crowd
Lets workers select responses from suggestions
Submit Response
Collaborative reasoning system
Rewards workers that support consistent conversation
Chat Worker
Dynamic scoring system
Browse working memory
Lets workers promote portions of a conversation to a shared memory
Curated memory system
Solo Worker
Give Feedback
Vote for important facts in the memory
Register
Memory Worker
User
Browse the conversation history
Send Message
Edit Working Memory
Send Photos

Figure 3.2.1: Use Case Diagram for Chorus System

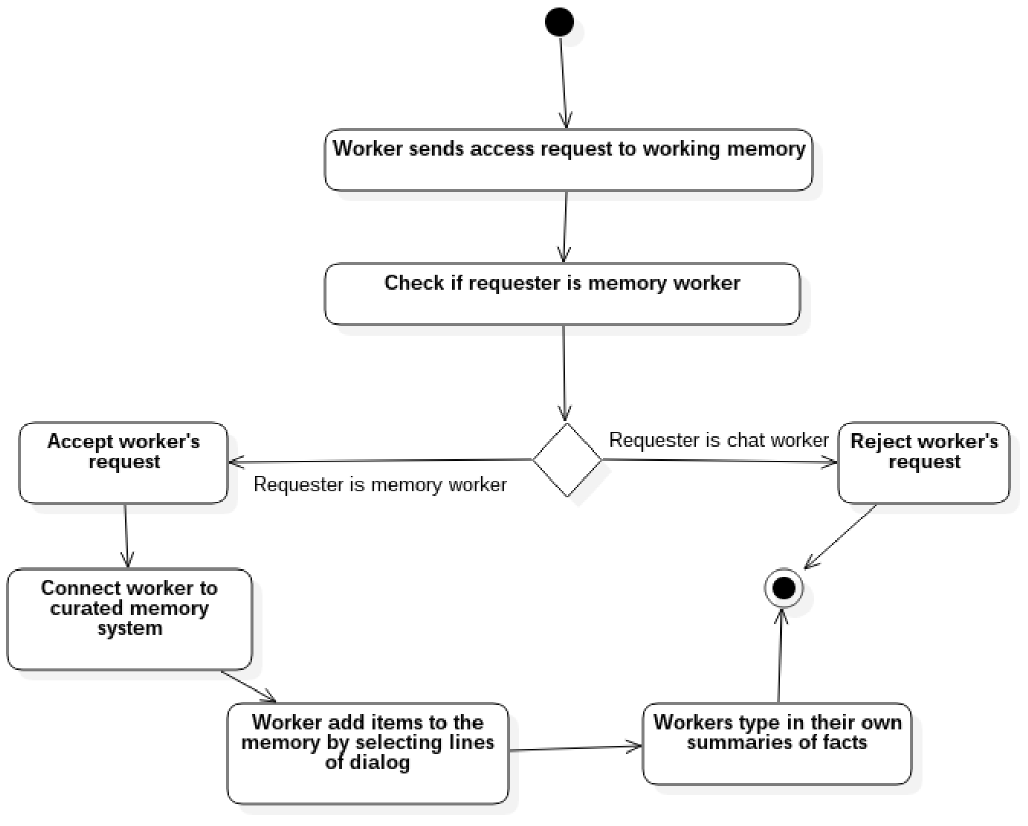
Use Case	Description
Provide access to Chorus	Google Hangouts give access to Chorus so that Chorus can communicate with user. Hangouts is the platform where Chorus chat with the users.
Lets workers select responses from suggestions	Chorus' collaborative reasoning system allows workers to select best response among the proposed messages. Later, top voted messages are forwarded to the user.
Rewards workers that support consistent conversation	Chorus' dynamic scoring system rewards workers that support consistent conversation in order to encourage workers to work harder & efficient.
Lets workers promote portions of a conversation to a shared memory	Chorus' curated memory system allows workers to add important facts to memory. So that, crowd can learn and remember information about user.
Give Feedback	User is given a feedback form after the conversation ends. User is expected to submit his/her feedback in 10 minutes. Otherwise the feedback is discarded.
Register	When a user wants to use the Chorus for the first time, he/she has to register to the system. This operation happens only once. Afterwards, Chorus looks on the Hangouts menu.
Send Message	User can send message via Google Hangouts in order to communicate with Chorus Bot.

Send Photos	User can send photos via a button in the chat menu. He/she also can draw something and send it.
Vote for proposed responses	Each worker vote for the proposed answers. Among them, only top rated answer is forwarded to user.
Submit Response	Workers can submit their responses. However, among submitted responses, the top rated response will be forwarded to user.
Browse working memory	Workers can browse working memory to get some facts about the user and submit their responses according to those facts. Data is being uploaded to memory with this way.
Vote for important facts in the memory	Memory workers can vote for facts in the memory they think are important. With that way, unnecessary data can be eliminated.
Browse the conversation history	Memory workers can browse conversation history and make additions to working memory from past conversations. So, system holds more information about the users.
Edit Working Memory	Memory workers can edit working memory by adding some lines from conversation with the user or by adding their own summaries of facts. So, system holds more relevant & necessary information about the user for the sake of a better chat with user.

Table 3.2.1: Use-Case Descriptions Table

Use Case Id	1
Use Case Name	Edit Working Memory
Trigger	Worker sends access request
Precondition	Requester must be a memory worker
Primary Actors	Memory worker, Curated Memory System
Description	Memory worker adds items to the memory or they type their own summaries of facts
Basic Path	1. Worker sends request 2. System checks for authentication 3. Memory worker adds items 4. System saves messages to memory
Exception Path	If not authenticated, system sends "Permission denied" reply.

Table 3.2.2: "Edit Working Memory" Table



Worker sends access request to working memory
Check if requester is memory worker
Requester is chat worker
Reject worker's request
Accept worker's request
Requester is memory worker
Connect worker to curated memory system
Worker add items to the memory by selecting lines of dialog
Workers type in their own summaries of facts

Figure 3.2.2 : Activity Diagram for "Edit Working Memory"

Use Case Id	2
Use Case Name	Give Feedback
Trigger	1. User sends message to end conversation 2. Time out
Precondition	Conservation must end
Primary Actors	User, Worker
Description	Chorus system sends a feedback form to user for evaluating workers
Basic Path	1. Conversation ends 2. System sends a feedback form 3. User gives feedback
Exception Path	User do not answer feedback form

Table 3.2.3: “Give Feedback” Table

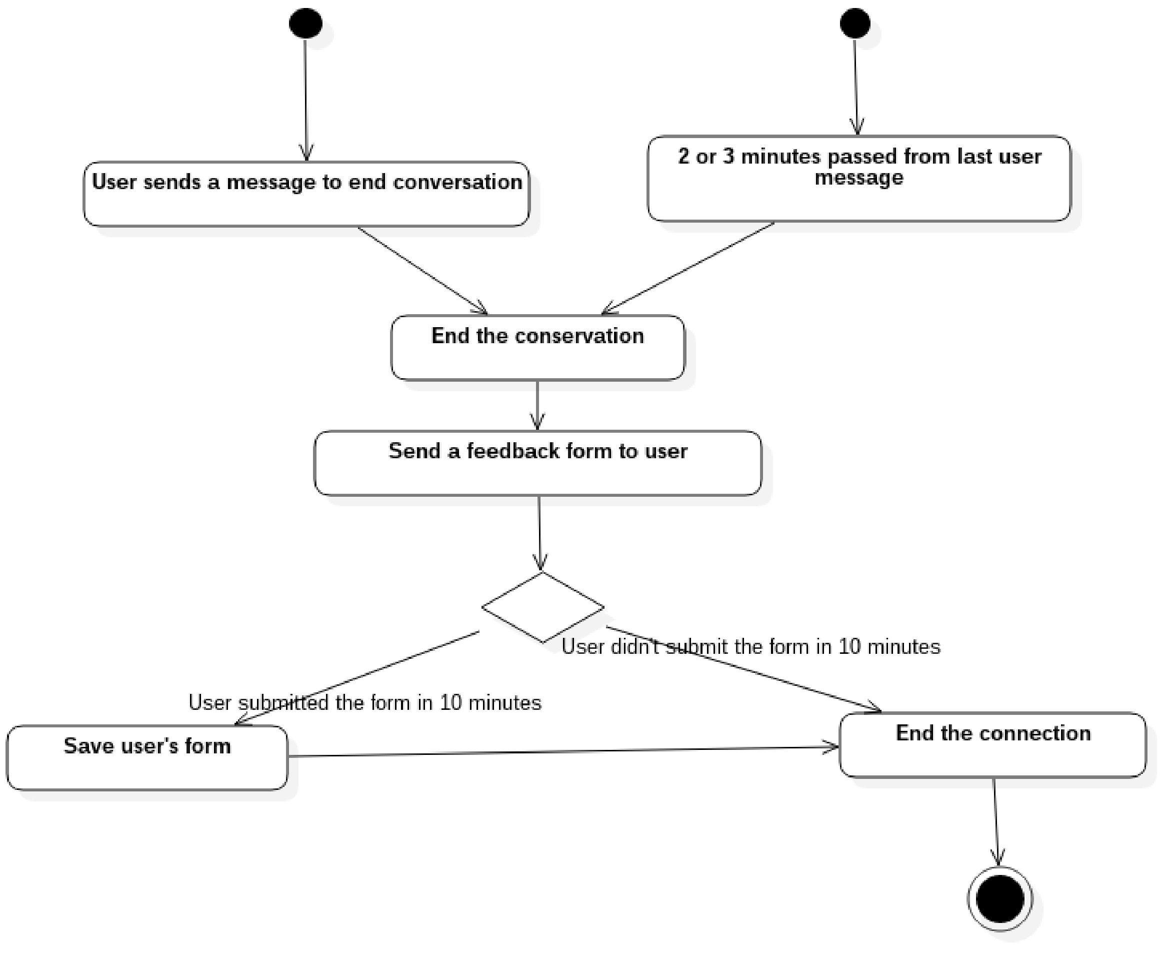

User sends a message to end conversation
2 or 3 minutes passed from last user message
End the conservation
Send a feedback form to user
User didn't submit the form in 10 minutes
User submitted the form in 10 minutes
Save user's form
End the connection

Figure 3.2.3 : Activity Diagram for “Give Feedback”

3.3. Usability Requirements

- Chorus shall work on an instant messaging system like Google Hangouts.
- At least one worker shall be available to respond to user at any time.
- Workers shall both submit responses and vote for proposed messages by other workers in the crowd.
- A user shall be able to evaluate the workers by giving feedback.
- A user must fill the form and sign up to use Chorus.

3.4. Performance Requirements

In order to provide a good experience to users, Chorus system should satisfy the followings:

- Chorus shall be able to communicate with at least 400 users simultaneously.
- Responding to user’s first message shall be within 2 minutes.
- Chorus shall provide accurate, logical responses at least 90% of user queries and stay on-topic in at least 90% of responses.

3.5. Logical Database Requirements

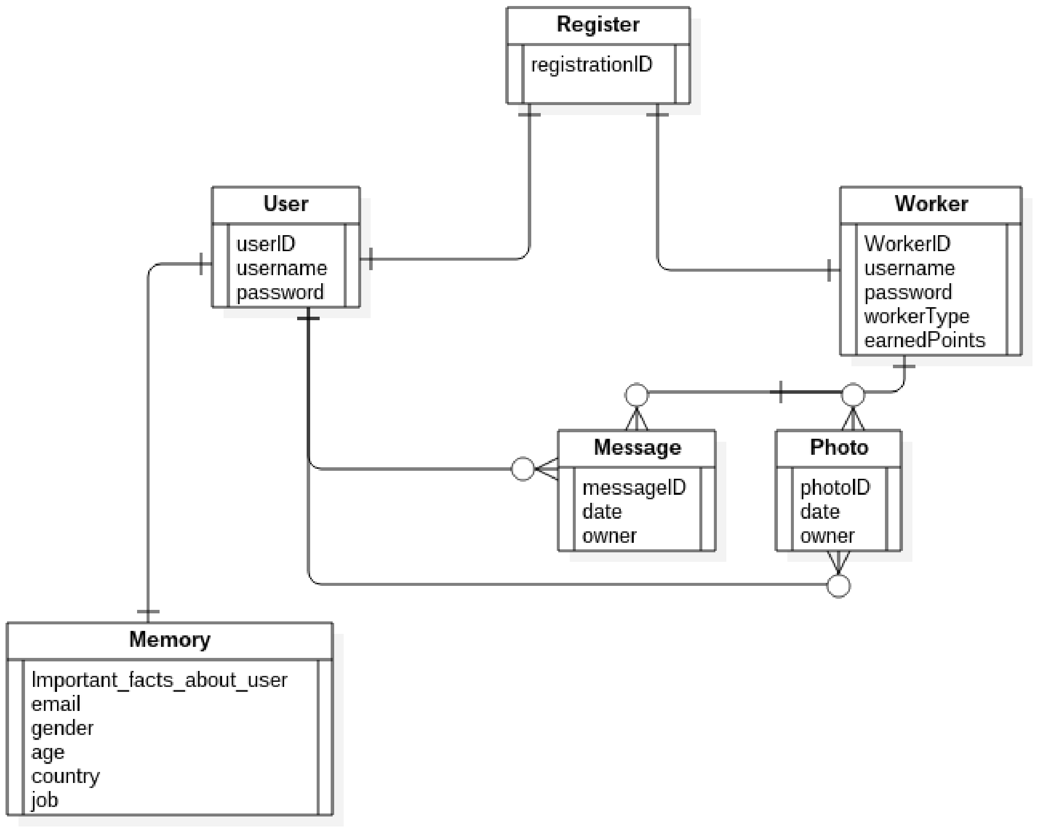


Figure 3.5.1: E-R Diagram for Chorus

Chorus' database system shall store all the users and workers' registration information, plus their ID, username and password. That is being checked at the beginning of the each conversation.

Workers' earned points shall be stored in order to pay them fairly.

Since every conversation must be stored, all the messages and photographs shall be stored with their date and sender. Each message sent by Chorus Bot or user is stored just after message sent. It is also important for legal issues.

Most important part of the database is Working Memory. Working Memory is being updated after each conversation. Here, all the important data is stored for each user, such as his/her budget, eating habits. This data improves the quality of conversations because responses go to user in accordance with his/her special enjoyments, needs etc.. This data is being collected from the conversation. And important facts about user added there by the memory workers after each conversation.

3.6. Design Constraints

- Chorus does not support video or audio calls for privacy concerns.
- All the conversations are stored for legal issues.
- All the messages must obey ethical rules, workers must not abuse users.

3.7. Software System Attributes

3.7.1. Reliability

- The crowd can answer with a high quality thanks to voting system. Workers types messages and agree with the best one. That makes Chorus reliable & trustable.
- Because backup is being uploaded to database after each conversation, there is no loss. In addition, Chorus works on Google Hangouts, that means Google backups conversation to user's account, so Chorus as reliable as Google Hangouts.

3.7.2. Availability

- Since the project is free and operates on Google Hangouts, anyone who has a gmail account and internet connection can use it easily.
- System is always open. It is available for 24 hours in a day, because there must be at least one worker at any time.

3.7.3. Security & Privacy

- Hangouts messages are encrypted over an HTTPS connection with 128-bit encryption and TLS 1.2 and since Chorus runs on Hangouts, it is secure.
- User must login gmail to use Chorus, so it's authentication checked by Google. Clearly, Chorus is as secure as Google Hangouts.
- Workers(MTurks) cannot even see user's name. They can only know what user tells. So, system is private.

3.8. Supporting information

Recently, crowd computing has become very popular to solve problems, it is more effective than autonomous software. In such a system, problems are solved by groups of paid human workers over the web. Multiple workers also falls the average task time from 103.4 seconds with a single worker to 44.6 seconds because multiple workers have a better chance of finding a solution faster. Chorus is a great example of such a system.

Chorus shows that conversational assistants (combining human and artificial intelligence) can be used in reality.

System works on Google Hangouts and looks as a random friend from Hangouts. System has a database which stores important data of users. The database is being used to communicate with users well.

Workers are getting paid by Chorus.

In terms of response speed, the first response from workers in a conversation took an average of 72.01 seconds

Experiments demonstrates that the crowd was collected from a combination of Amazon MTurks (77% of workers) and MobileWorks(23%).