SOFTWARE DESIGN DESCRIPTION

For Chorus Project

Bilal Özlü 1942614

Ali Şimşek 2099752

Version 1.1 (27.04.2018)

Table of Contents

List of Figures

List of Tables

1.Introduction

1.1. Purpose of the System

The purpose of this SDD document is to give necessary information about the design of the Chorus project and how it is expected to build with its components. This document must be followed for the understanding and maintenance of the system.

1.2. Scope

This SDD document describes Chorus system at the design level in accordance with the Software Design Description of IEEE Standard 1016-2009. This document includes some UML diagrams(Component diagram, Deployment diagram etc.) to clearly show how the system and its components designed and how the system interacts with external world.

1.3. Stakeholders and their concerns

- **Users**

  Users are the people who register with their Gmail account and use Chorus via Google Hangouts. They want Chorus Bot to be always available. Their main concern is getting practical and logical advice from the Bot. Also, they want to get quick response. Another important concern of users is their privacy, they do not want to give private information about themselves and they do not want the sharing of the information they provided. In addition, users may want to send photos to illustrate what they mean in the chat.

- **Chat Workers**

  Chat Workers (Amazon MTurks) are the paid human workers who replies users’ messages. They want to search and see important facts about the user while chatting. They want working memory to be unambi. They also want fair payment.

- **Memory Workers**

  Memory Workers are responsible from the working memory. They browse the conversations and add important facts to memory. They want conversations to be understandable and informative.

- **Developers**

System developers are programmers who develop the software of Chorus. Since they want steps of the system to be obvious and clear, they need well designed SRS and SDD documents. They also want to develop the software in a simple and comfortable environment.

- **Maintainers**

System maintainers are the IT staff who handle with the possible errors and maintain the system. They want code to be clean and modular to understand it easily.

2. References

- IEEE Standard for Information Technology—Systems Design—Software Design Descriptions https://standards.ieee.org/findstds/standard/1016-2009.html
- Chorus: A Crowd-Powered Conversational Assistant by Lasecki https://dl.acm.org/citation.cfm?id=2502057
- Website of Chorus https://talkingtothecrowd.org/

3. Glossary

Term	Definition
User	People who register with their Gmail account and use Chorus
Chat Worker	Chat Workers (Amazon MTurks) are the paid human workers who replies users’ messages
Memory Worker	Memory Workers browse the conversations and add important facts to working memory.
Working Memory	A memory consist of users’ important facts that might be useful when replying to user.
HTML5	HyperText Markup Language, it is used to create webpages. HTML5 is the latest version.
CSS	Cascading Style Sheets, it is used with HTML to design webpages.
Bootstrap	Open source front-end library for designing webpages.
MySQL	Open source relational database management system.
phpMyadmin	Open source administration tool for MySQL.
JVM	Java virtual machine, it allows computers to run Java.

Table 3.1: Glossary

4. Architectural Views

4.1. Context View

This part of the document explains Chorus system’s interactions with user. (System components and the relationship between those components). We used a Use Case Diagram to illustrate the relationships between Chorus and external systems.

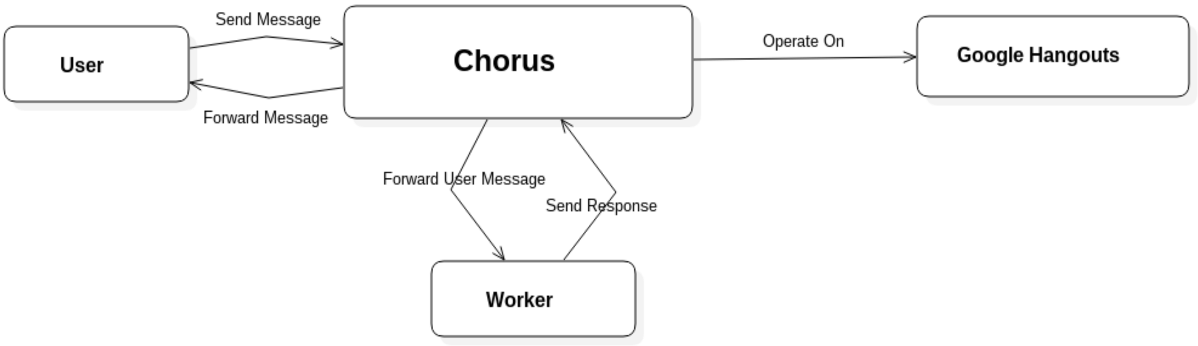


Figure 4.1.1: Context Diagram for Chorus System

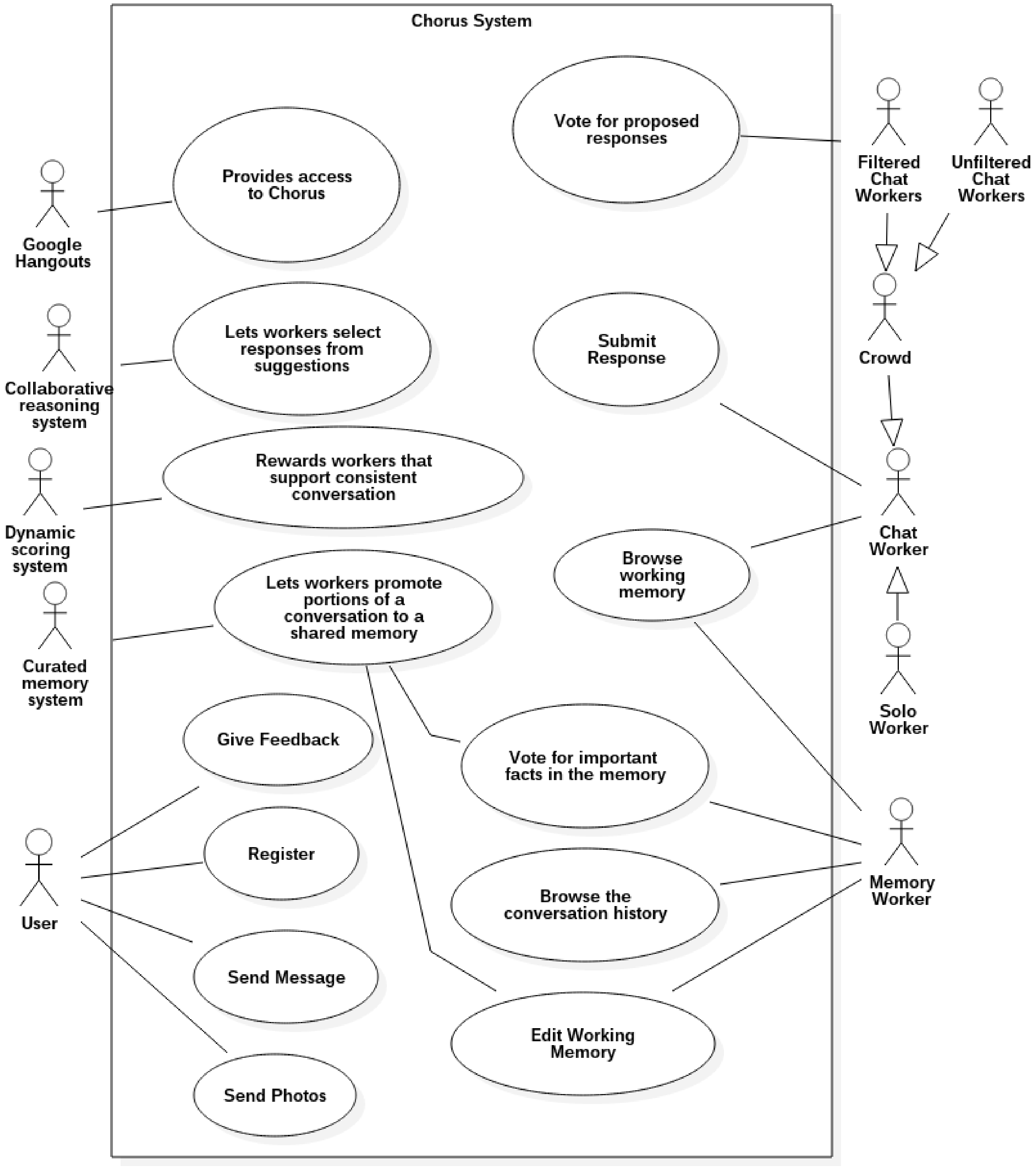

Chorus System
Vote for proposed responses
Provides access to Chorus
Lets workers select responses from suggestions
Submit Response
Rewards workers that support consistent conversation
Browse working memory
Lets workers promote portions of a conversation to a shared memory
Give Feedback
Vote for important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

Figure 4.1.2: 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 4.1.1: Use-Case Descriptions Table

4.2. Composition View

This part of the SDD document is about the components of the Chorus Project. Component and Deployment diagrams are shown below.

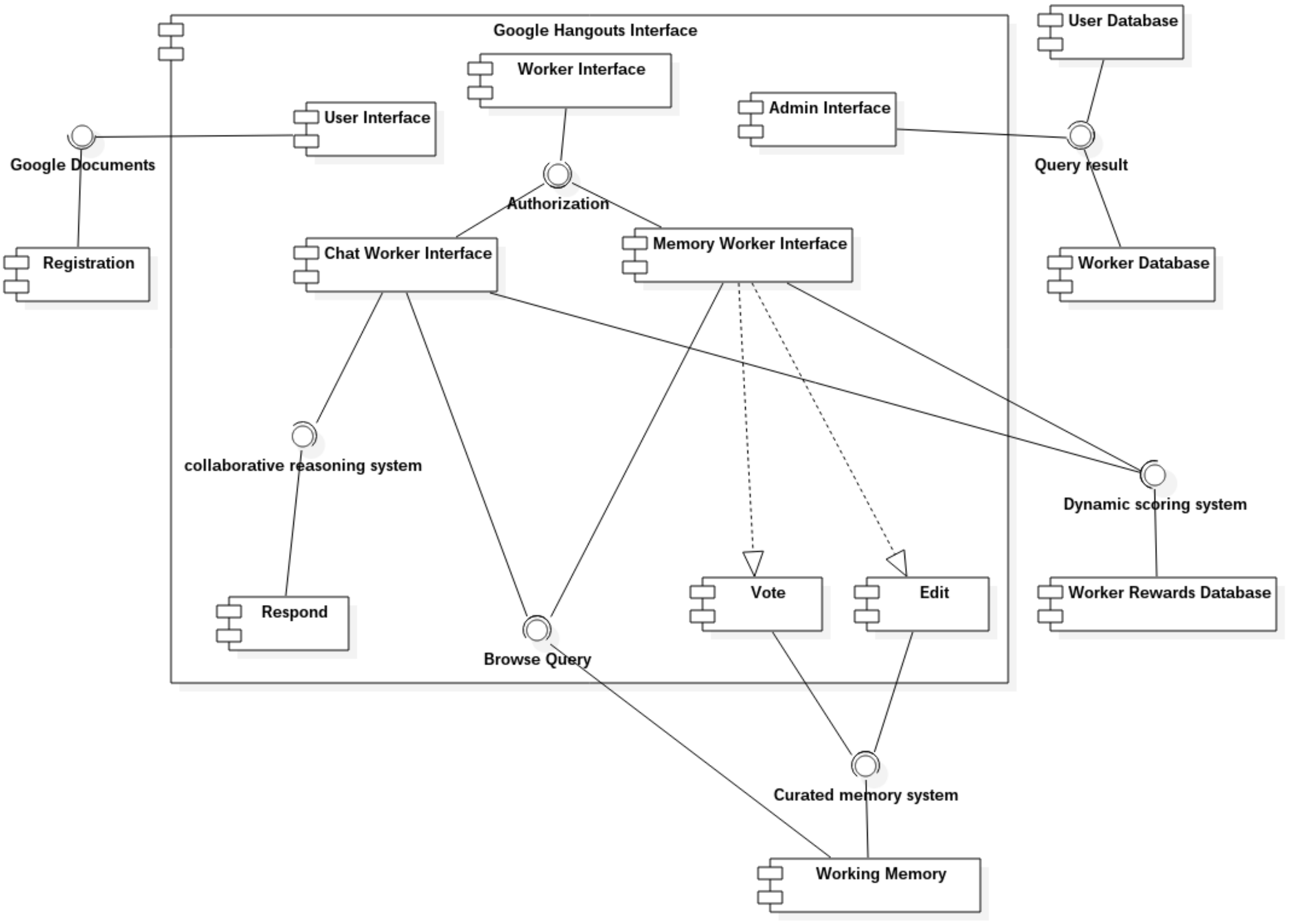


Figure 4.2.1: Component Diagram of Chorus

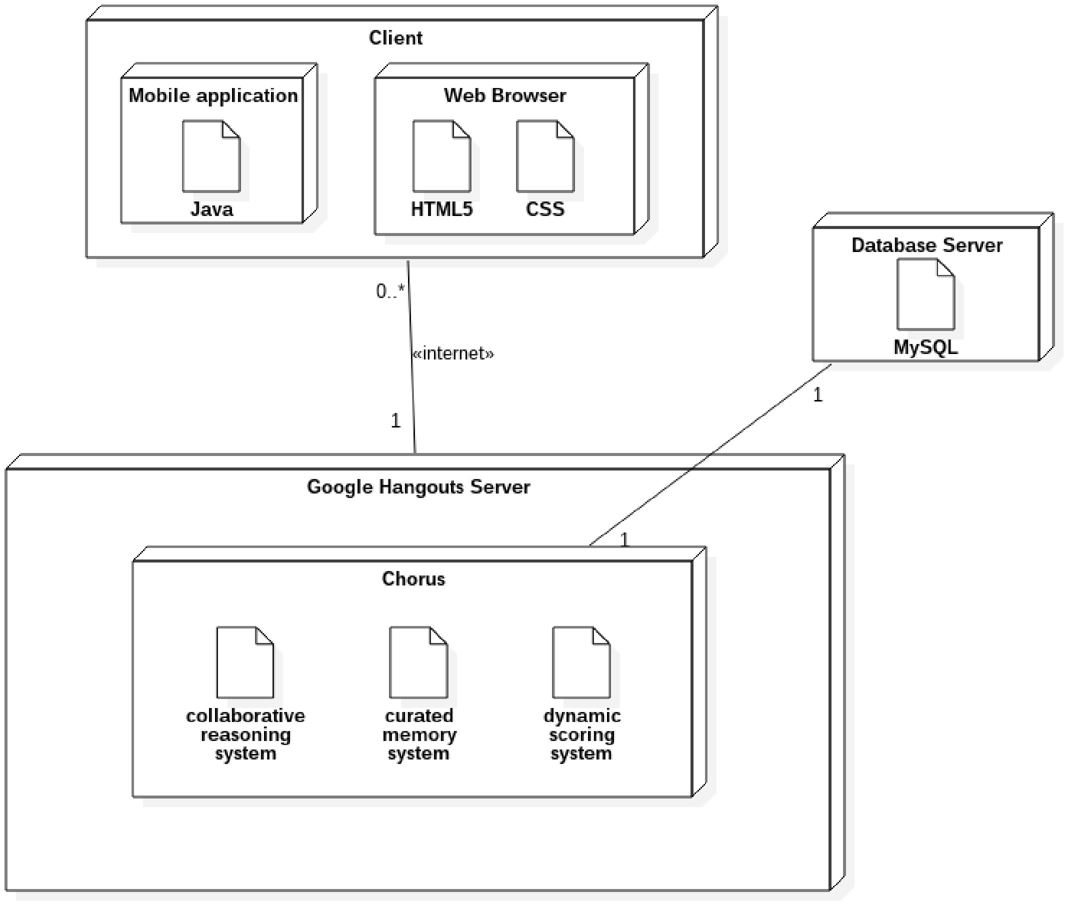


Figure 4.2.2: Deployment Diagram of Chorus

4.2.1. Design Rationale of Composition View

- Chorus use Hangouts in order not to spend time&energy on developing a new secure messaging system.
- There is a dynamic scoring system to pay workers fairly.
- We used Java for mobile application to make it more portable and more secure. JVM provides extra security layer for the application and it also enables us to use the same code on different systems which makes application more portable.
- Our system use MySQL as database system since it is open source and widely used.
- We used CSS on our browser application because its easy to maintain.By making one change to the website's CSS, elements in all the web pages will be updated automatically

- We used HTML5 because it enables designers to use cleaner and understandable code. Div tags can be replaced with some HTML5 elements.
- To make system more reliable, we separated the databases. (as User Database and Worker Database). Even if one database encounters with an error, others will work.

4.3. Information View

In this section of the document, class diagram and entity-relationship diagram are used to illustrate the main structure of the project. Persistent data stored in the database is shown as classes and the relations between those classes are indicated by arrows.

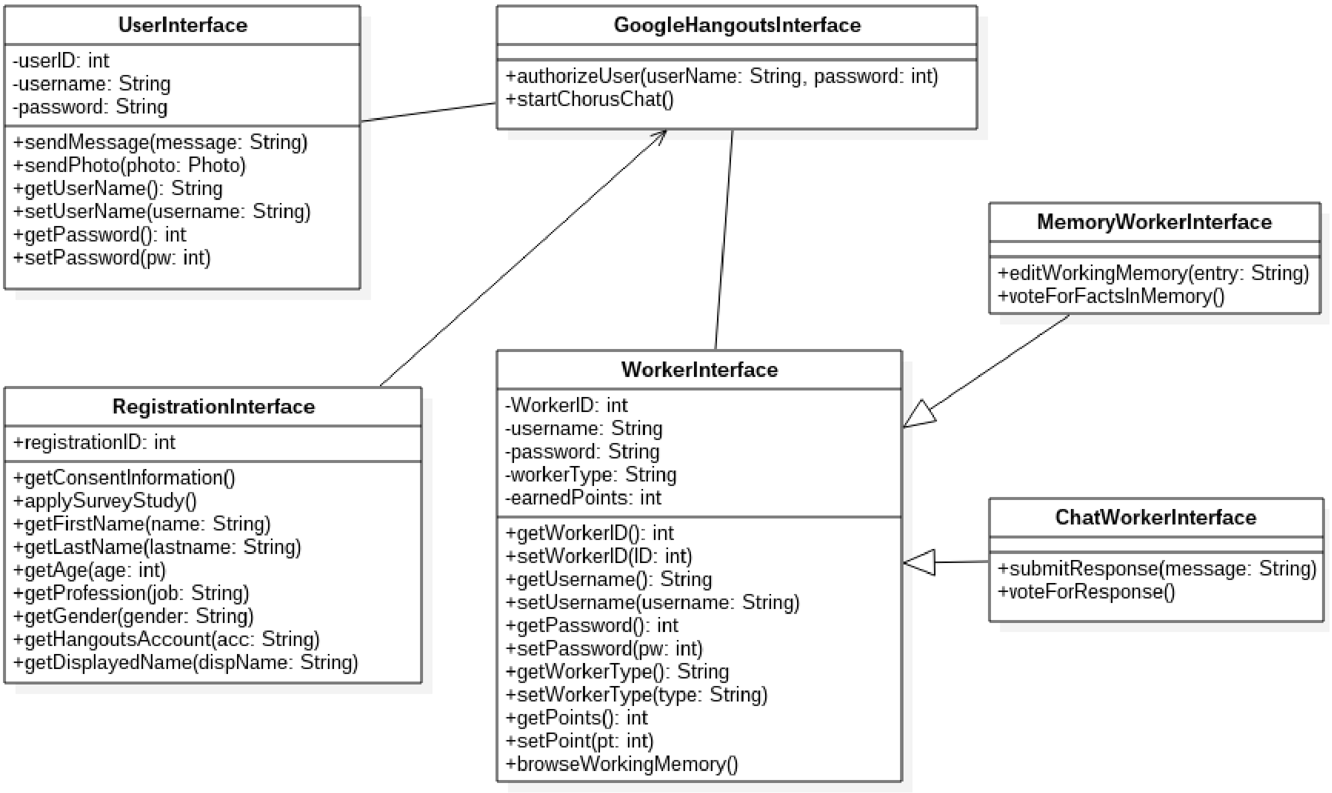


Figure 4.3.1: Class Diagram of Chorus

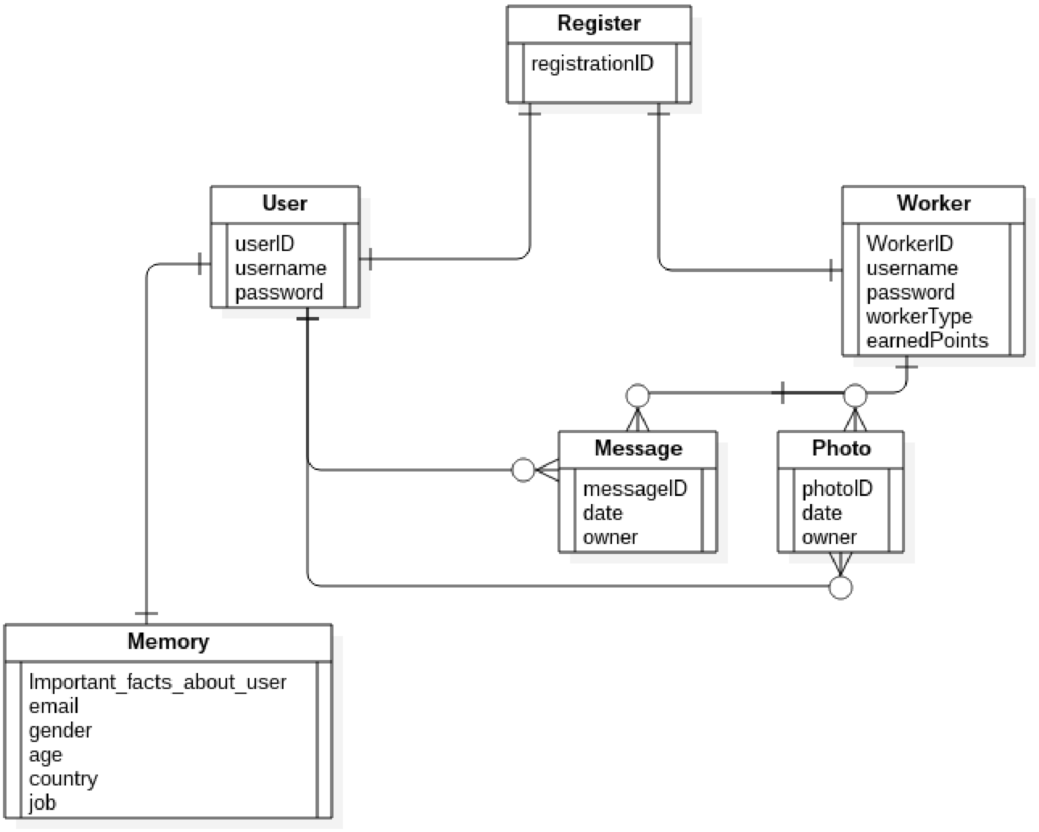


Figure 4.3.2: E-R Diagram for Chorus

Operation	**CRUD**
sendMessage	Create: Message Read: - Update: - Delete: -
sendPhoto	Create: Photo Read: - Update: - Delete: -
getHangoutsAccount	Create: Register Read: - Update: - Delete: -

editWorkingMemory	Create: - Read: Memory Update: Memory Delete: Memory
voteForFactsInMemory	Create: - Read: Memory Update: Memory Delete: -
submitResponse	Create: Message Read: - Update: - Delete: -
voteForResponse	Create: - Read: Message Update: - Delete: -
browseWorkingMemory	Create: - Read: Memory Update: - Delete: -

Table 4.1.1: CRUD Operations Table

4.3.1. Design Rationale of Information View

- The system's database designed as two different database(worker database and user database) for the sake of a more reliable system. In user database, we stored all the information about the user(messages, important facts etc.). In worker database, we stored the workers' information(messages, earned points etc.).
- Those design choices are made for better security and maintainability. All properties can be updated according to stakeholders' needs.

- Users are unable to reach the worker database for privacy reasons. And also, workers cannot see the users' data unless they are authorized to see and use it just for the sake of a better conversation.
- We used MySQL on phpMyAdmin for relational database management because it meets all of our needs about database operations, also it is very common and reliable.

4.4. Interface View

This part of the document demonstrates internal and external interfaces of the Chorus project. Previously, we showed the service interfaces as a class diagram in Figure 4.3.1. Below, some sequence diagrams are provided in order to illustrate interface design. In addition, the logic behind the design is explained in the design rationale part.

4.4.1. Internal Interface

Google Hangouts provides a chatting platform to user. User chats with the Chorus Bot via Hangouts. When user requests to communicate with Chorus Bot, his/her authorization is checked by Gmail. User has to have a Gmail account in order to register and use Chorus. User can send messages/photos to bot and those messages/photos are taken by Chorus, stored in the User database of Chorus system. Some important facts are chosen from user's messages by the memory workers.

Workers also use Google Hangouts to chat with user, also some workers reach the working memory if they are authorized. First, workers enter the sytem with their username and password. In authorization, they are divided according to their worker type. Chat workers chat with users via Hangouts and they can reach some important facts about users, they can see the information that Chorus allows them to access, otherwise they cannot know even name of the user. On the other hand, memory workers access to working memory. They edit and vote for the important facts about the user. Also, they queires the database to choose these facts for each user.

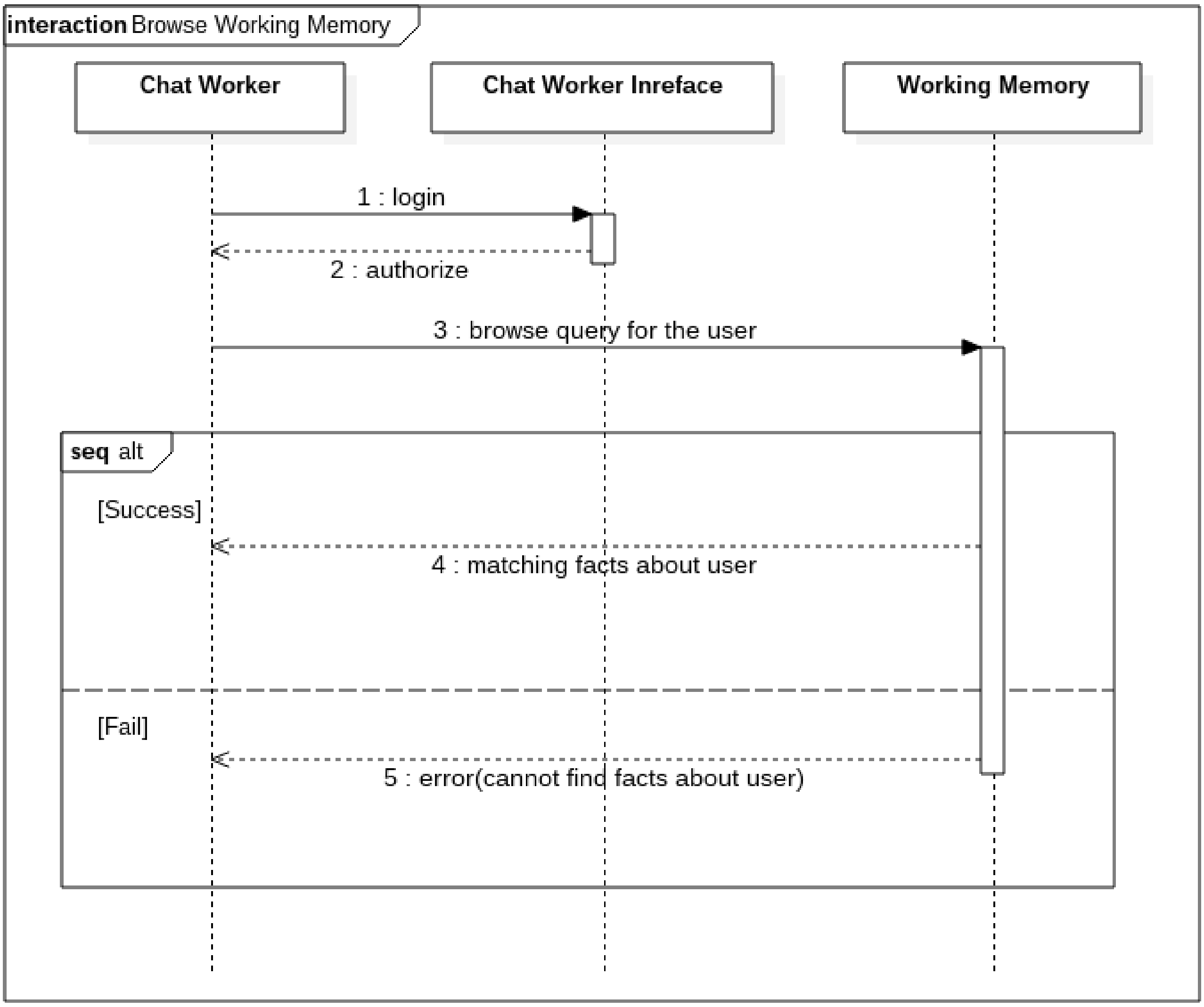
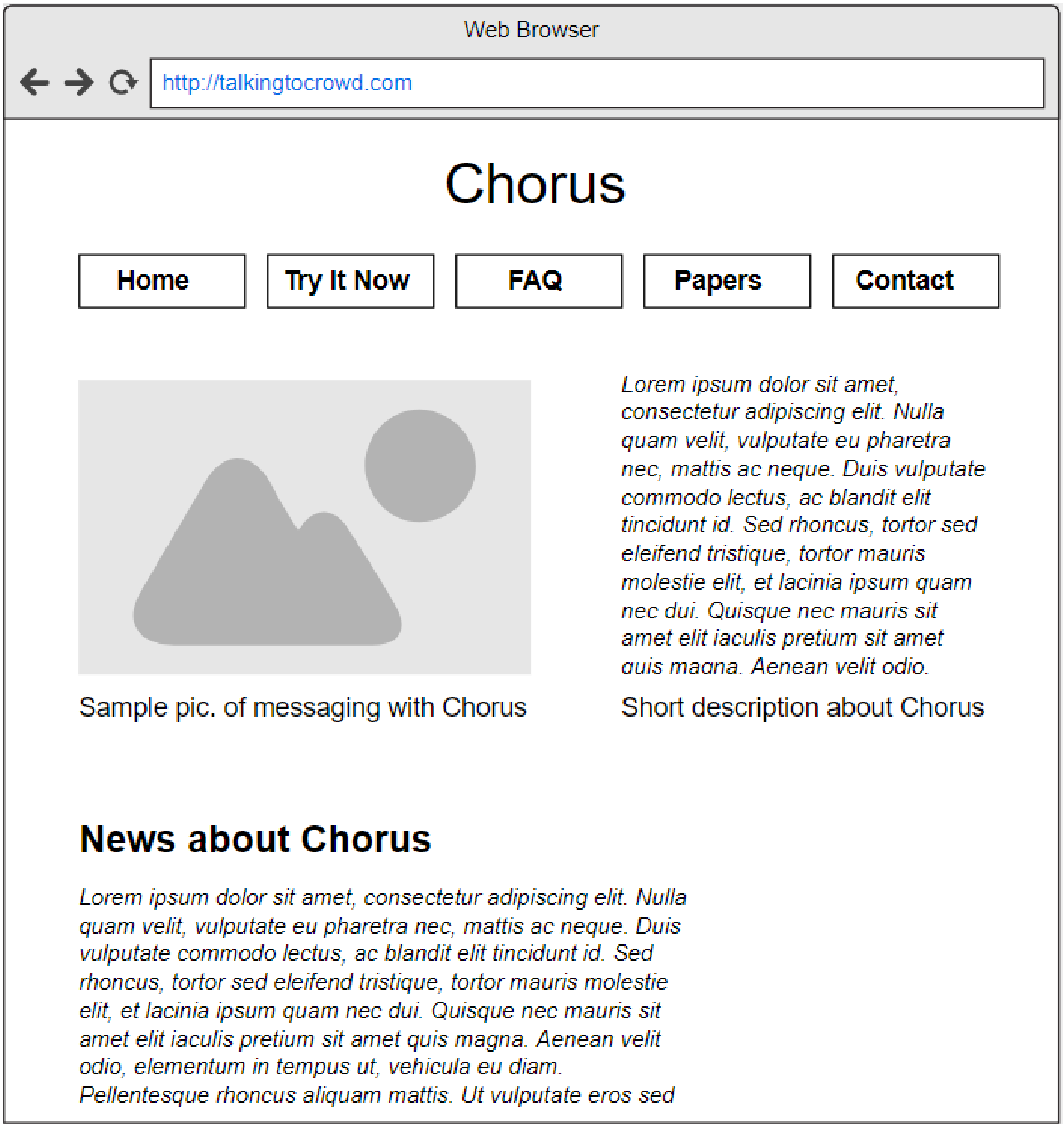

interaction Browse Working Memory
Chat Worker
Chat Worker Inreface
Working Memory
1 : login
2 : authorize
3 : browse query for the user
seq alt
[Success]
4 : matching facts about user
[Fail]
5 : error(cannot find facts about user)

Figure 4.4.1.1: Sequence Diagram for Browse Working Memory (internal interface)

4.4.2. External Interface

Chorus has a webpage (named as talkingtocrowd.com) for introducing Chorus and registering to Chorus. In the webpage, there is a “Try It Now” button to register Chorus, it can be seen in Figure 4.4.2.1.



Chorus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio.

Sample pic. of messaging with Chorus

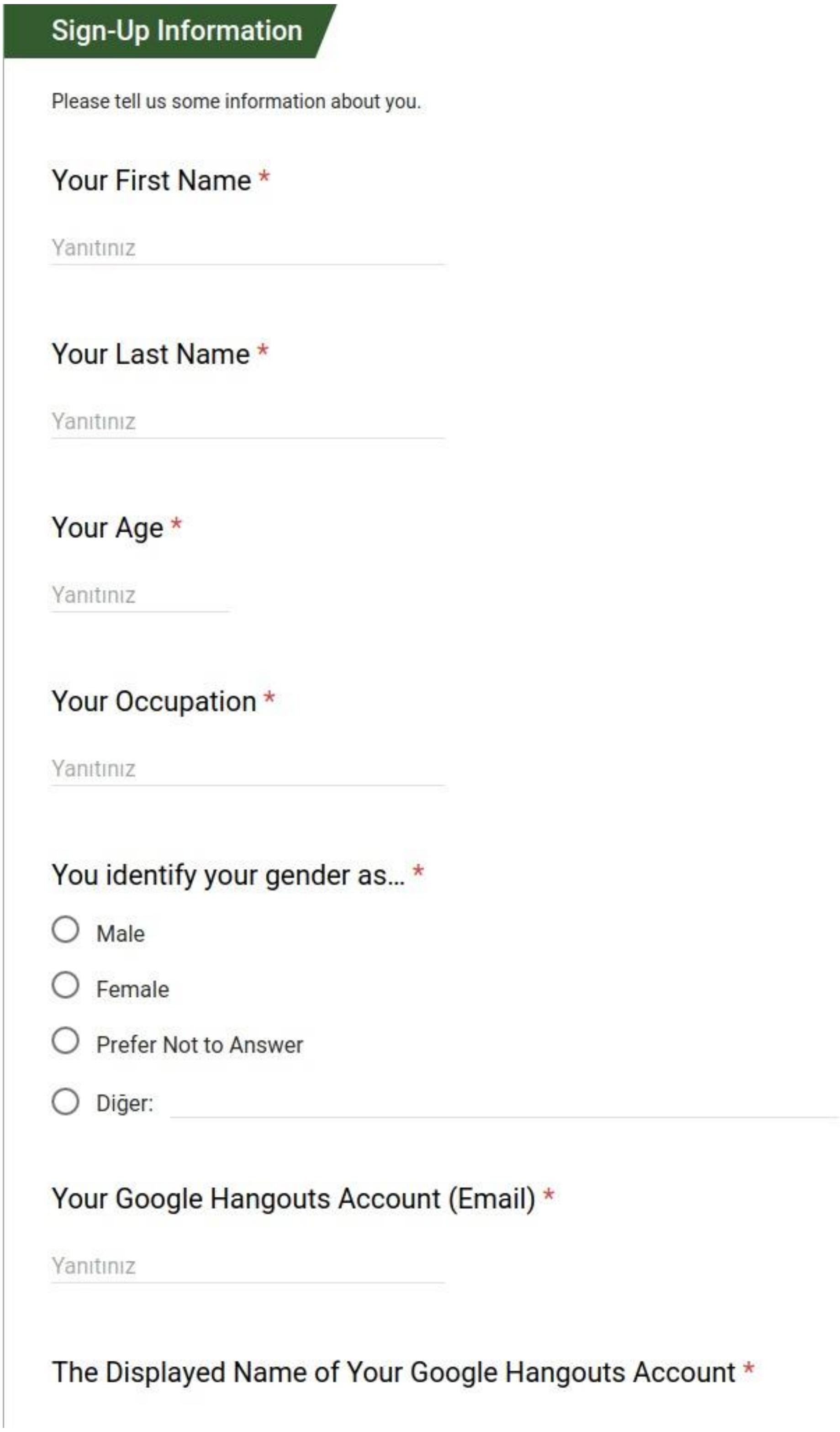
Short description about Chorus

News about Chorus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed

Figure 4.4.2.1: Webpage of the talkingtocrowd.com

“Try It Now” link directs to a Google form that gives a way to register and join as a user to Chorus. That Google Form(Figure 4.4.2.2) is the registration interface of Chorus. It must be filled to register.



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) *

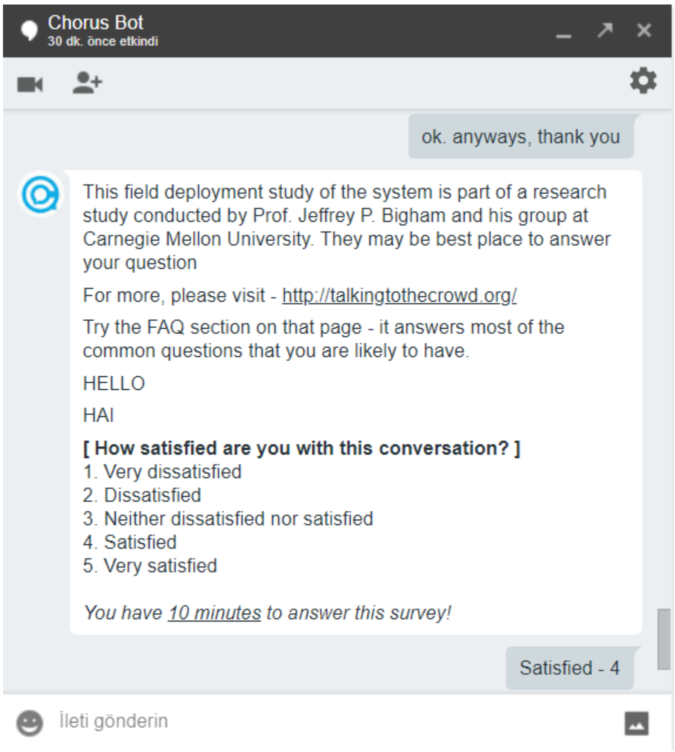
Yanıtınız

The Displayed Name of Your Google Hangouts Account *

Figure 4.4.2.2: Registration form of the Chorus

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 4.4.2.3).



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

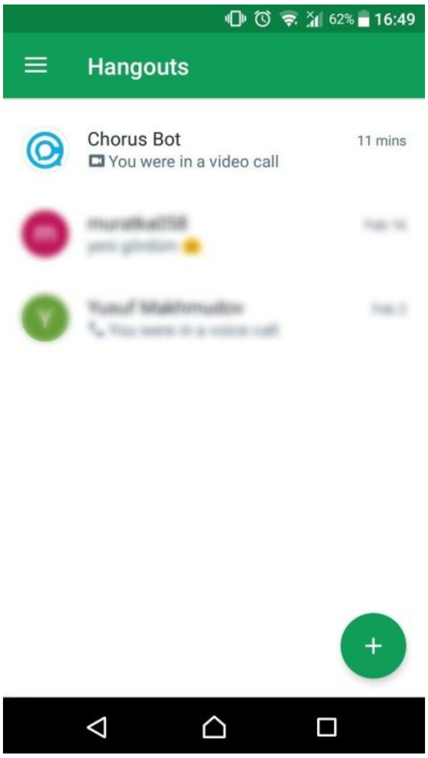
[How satisfied are you with this conversation?]
1. Very dissatisfied
2. Dissatisfied
3. Neither dissatisfied nor satisfied
4. Satisfied
5. Very satisfied

You have 10 minutes to answer this survey!

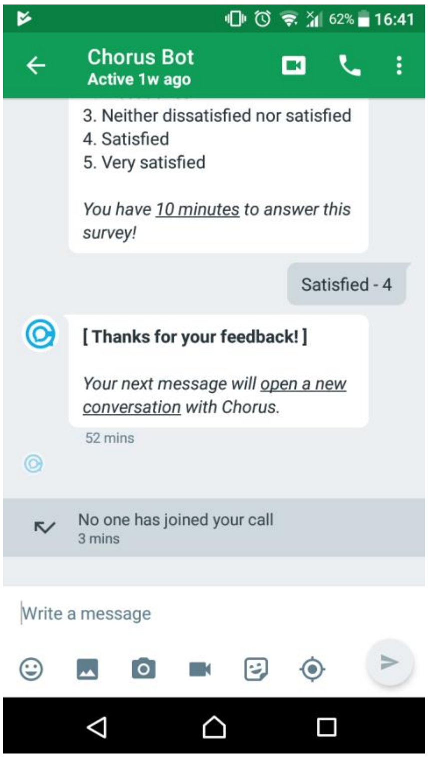
Satisfied - 4

Figure 4.4.2.3: Chorus chat window

Here are screenshots of Chorus on mobile devices. From the Figure 4.4.2.4, we can see that Chorus Bot is just another contact in Hangouts' contact list.



Chorus Bot 11 mins
You were in a video call



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 4.4.2.4: Mobile1

Figure 4.4.2.5: Mobile2

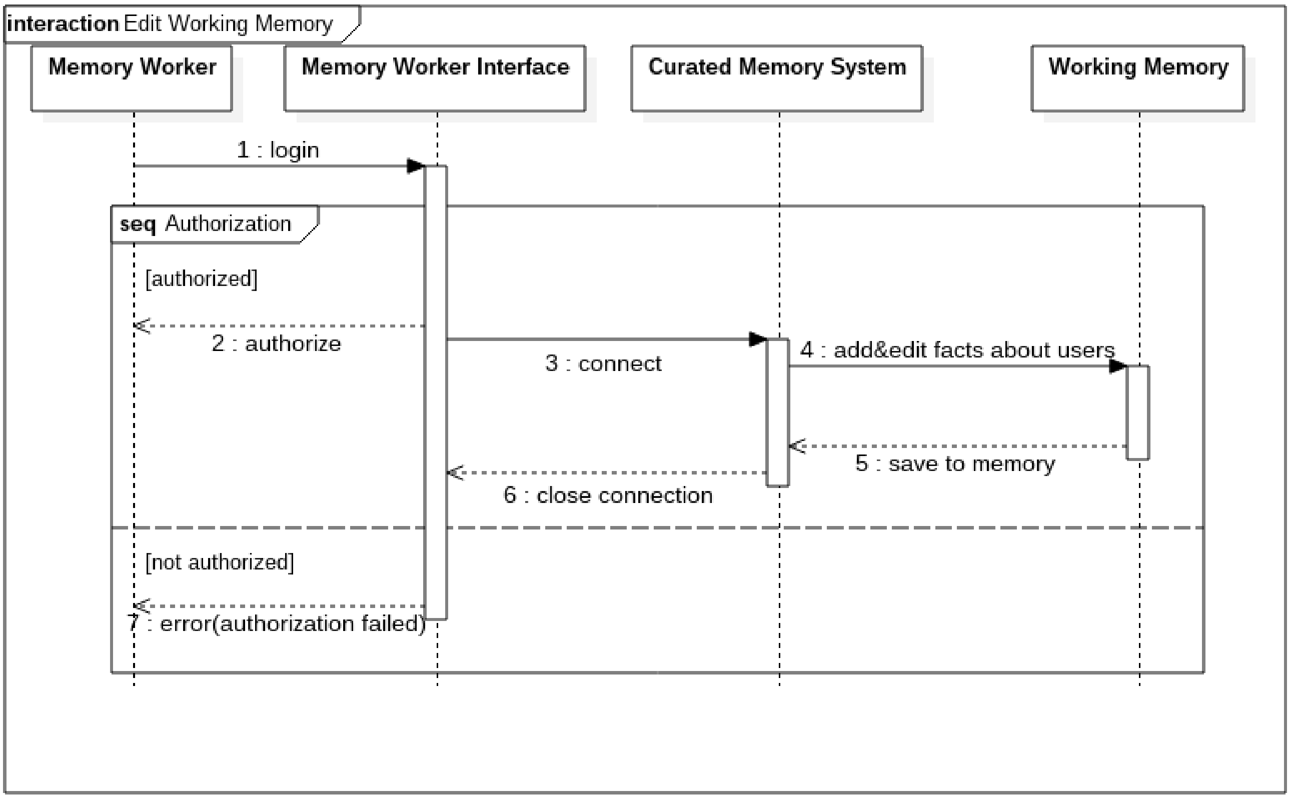


Figure 4.4.2.6: Sequence Diagram for Edit Working Memory (external interface1)

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 or edits items 4. System saves messages to memory
Exception Path	If not authenticated, system sends “authorization failed” reply.

Table 4.4.2.1: “Edit Working Memory” Table

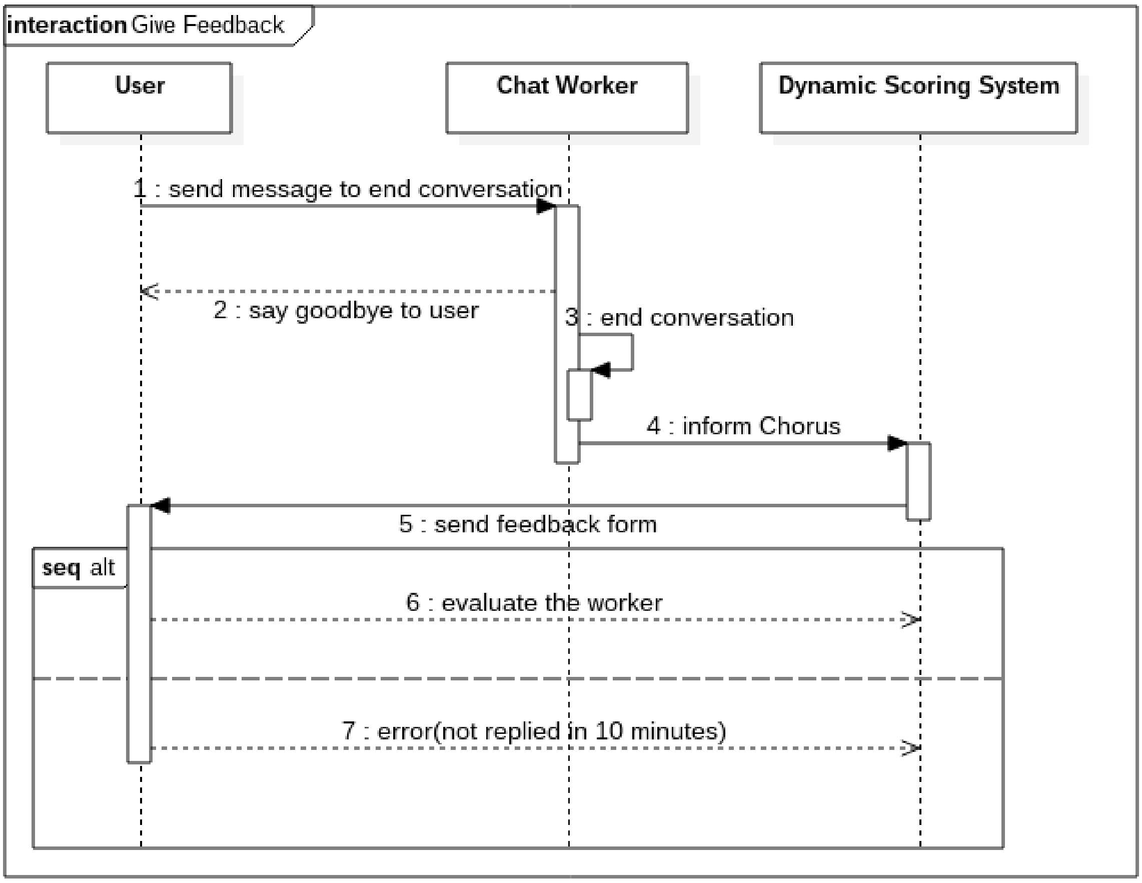


Figure 4.4.2.7: Sequence Diagram for Give Feedback (external interface2)

Use Case Name	Give Feedback
Trigger	User sends message to end conversation
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 4.4.2.2: “Give Feedback” Table

4.4.3. Design Rationale of Interface View

- While designing the webpage of Chorus (talkingtocrowd.com), we cared about simplicity. Everything is reachable from only one page.
- Bootsrap library is used in order to make website responsive to all devices. So, it can suit to all screen sizes.
- The registration form gets all necessary information by dividing the data into logical categories to put them into memory orderly.
- Since worker interface is just used by workers, the aesthetic details are not cared there.
- To encourage people to use Chorus, a familiar and well known chatting platform(Hangouts) is chosen . Most of the user interfaces drived by Google Hangouts because all the chats are done on it.
- For privacy reasons Chorus does not support video calls.
- To protect users’ privacy chat workers cannot see users’ name or mail account.