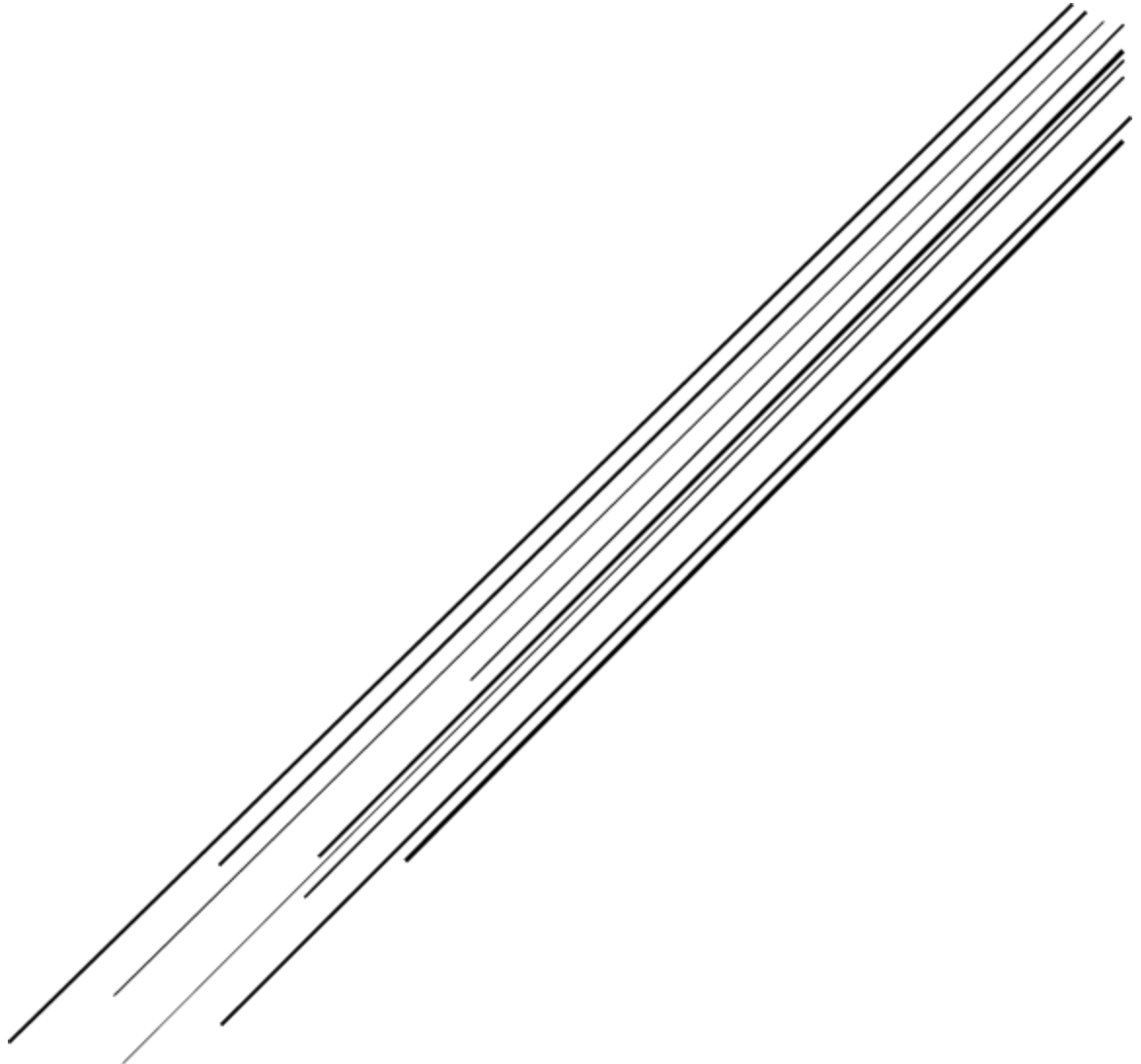


Software Requirements Specification

Chorus Project

Group-3

Yavuz Selim YEŞİLYURT	2259166
Ahmet Dara VEFA	2237899



12/03/2017

Table of Contents

List of Figures	2
1. Introduction	3
1.1. Purpose	3
1.2. Scope	3
1.3. Product Overview	3
1.3.1. Product Perspective	3
1.3.1.1. System Interfaces	4
1.3.1.2. User Interface	6
1.3.1.3. Hardware Interface	12
1.3.1.4. Software Interface	12
1.3.1.5. Communication Interface	12
1.3.1.6. Memory Constraints	12
1.3.2. User Characteristics	12
1.3.3. Limitations	13
1.4. Definitions	14
2. References	14
3. Specific Requirements	15
3.1. External Interfaces	15
3.1.1. Google Hangouts Interface	16
3.1.2. Google Interface	16
3.1.3. MTurk Interface	17
3.1.4. Amazon SimpleDB Interface	17
3.2. Functions	18
3.2.1. End User	18
3.2.2. Workers	27
3.2.3. Researcher	35
3.2.4. Administrator	40
3.3. Usability Requirements	49
3.4. Performance requirements	49
3.5. Logical database requirements	50
3.6. Design constraints	50
3.7. Software system attributes	51

List of Figures

Figure 1:	Context Diagram
Figure 2:	Main Screen For End User
Figure 3:	Sign Up Screen For End User
Figure 4:	Sign In Screen For End User
Figure 5:	Android Main Screen For End User
Figure 6:	Main Screen For Worker
Figure 7:	Main Screen For Admin
Figure 8:	A Sample Data Result For Researcher
Figure 9:	External Interfaces Class Diagram
Figure 10:	End User Use Case Model
Figure 11:	End User Sign In Sequence Diagram
Figure 12:	End User Sign Up Sequence Diagram
Figure 13:	Worker Use Case Model
Figure 14:	Worker Sign In Sequence Diagram
Figure 15:	Worker Sign Up Sequence Diagram
Figure 16:	Researcher Use Case Model
Figure 17:	Researcher Sign In Sequence Diagram
Figure 18:	Worker Sign Up Sequence Diagram
Figure 19:	Administrator Use Case Model
Figure 20:	Administrator Sign In Sequence Diagram
Figure 21:	Administrator Sign Up Sequence Diagram

1. Introduction

1.1. Purpose

Chorus Project is designed as a crowd-powered conversational assistant that allows a user to receive assistance on any online task through a two-way natural language conversation. Purpose of this project is to provide answers to requesters in real-time with enabling a conversation.

1.2. Scope

- The system will contain conversation servers to process the utterance created by end users and workers.
- The system will contain a database inside servers where all the information about workers and some past information about end users will be kept which enables new workers to join the conversation without any handicaps.
- The system will be designed to provide correct answers to users quickly.
- The system will have a voting mechanism for workers' responses, which will provide Real-Time Human Computation for the given answers to users.
- The system will check if the answer provided is sufficient enough continuously.
- The system will have a rewarding mechanism for the workers in accordance with their performance.
- The system will have a penalizing mechanism for the workers who responds with inappropriate languages and/or malicious words. This mechanism will rely on the users' and other workers' feedbacks.
- Providing the chat interface and creating new accounts, managing these accounts is not in the scope of this system.
- Also payment for the workers is not in the scope of this system.

1.3. Product Overview

1.3.1. Product Perspective

Chorus system is not exactly a part of a larger system but has other systems that it frequently interacts with, namely Google Hangouts, Google and MTurk. While Chorus servers and database will provide the structure for all the data transmission and will keep the information about users and staff, Google will be used to handle account management & authorization

processes and Google Hangouts, which is a subsystem of Google, will provide a chatting interface between user and the workers. Since the Google system is responsible for carrying out all of the user-management-related functionalities of Chorus system, two systems can be said to be interleaved.

On the other hand MTurk will be used as a retainer pool for recruiting workers and it will be responsible for the payment transactions between worker and the suppliers. Following context diagram represents interfaces of the project.

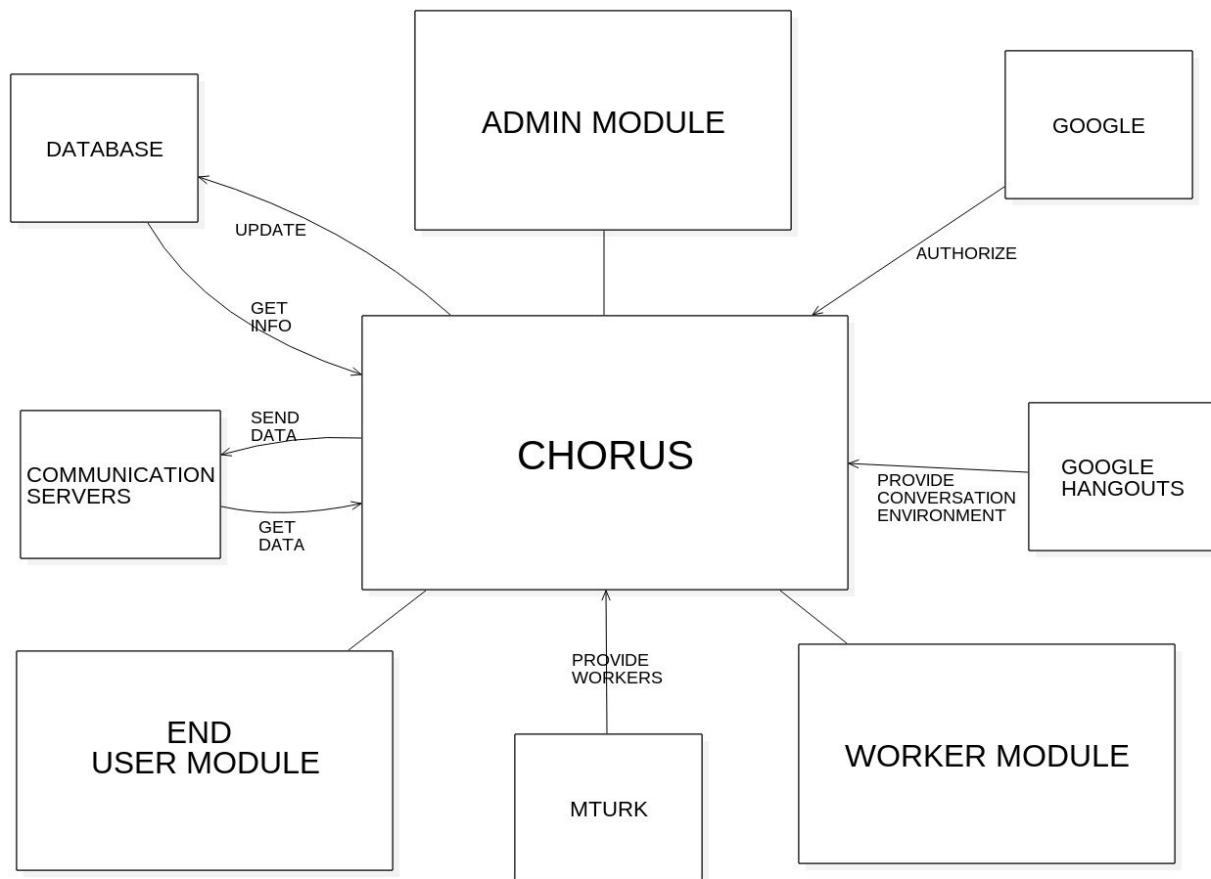


Figure 1: Context Diagram

1.3.1.1. System Interfaces

In Chorus project interfaces are divided into two parts as internal and external interfaces. Internal Interfaces are as the following: End User Module, Worker Module and Administrator Module. The external interfaces in the project are Database Management, Conversation Server Management, Google and Google Hangouts.

End User module consists of all the features that user can do, which are:

- Signing up to the Chorus system
- Signing in to the Google system
- Signing out of the Google system
- Starting a conversation with Chorus
- Viewing information page about Chorus
- Viewing FAQ page about Chorus
- Reviewing past conversation
- Archiving conversation
- Deleting conversation
- Reporting technical issues to the Google system
- Reporting technical issues to the Chorus system
- Deleting account

Worker Module consists of all the features that a worker can do, which are:

- Signing up to the Chorus system
- Signing in to the Chorus system
- Signing out of the Chorus system
- View past relevant data about user
- Updating data about user
- Proposing answer for user
- Voting for the best answer to be sent to user
- Reporting technical issues to the Chorus system

Admin Module consists of some set of features that an admin can do and some set of features that a researcher can do, admin features are:

- Signing up to the Chorus system
- Signing in to the Chorus system
- Signing out of the Chorus system
- Managing databases
- Managing servers
- Viewing reports
- Banning users
- Penalizing workers
- Rewarding workers

Researcher features are:

- Signing up to the Chorus system
- Signing in to the Chorus system
- Signing out of the Chorus system

- Viewing data results
- Evaluating data

Above items are module interfaces. In system interface, there are other structures as well. External interfaces are Google for the authorization and account management processes, Google Hangouts for the user interface, Amazon SimpleDB for building and maintaining database systems and frameworks for the development and MTurk for handling worker-related issues.

1.3.1.2. User Interface

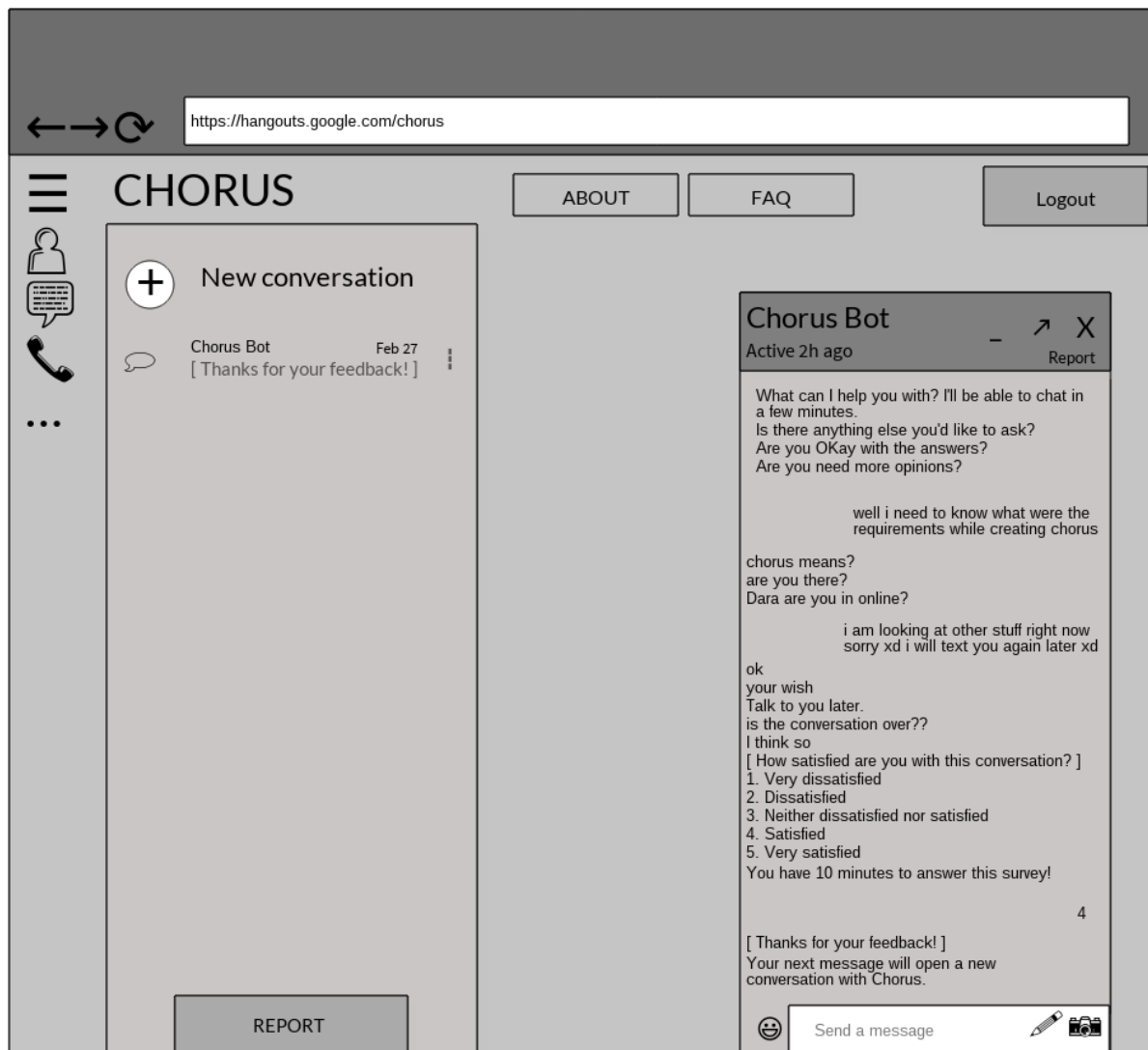


Figure 2: Main Screen For End User

End users interface for both browser and smartphone parts are entirely on Google Hangouts. For the browser interface, on the main page there is a sub window on the left (contact window) which shows a list of users of Google Hangouts including Chorus Bot as small subsections. End Users can archive or delete the past conversations they had with Chorus Bot

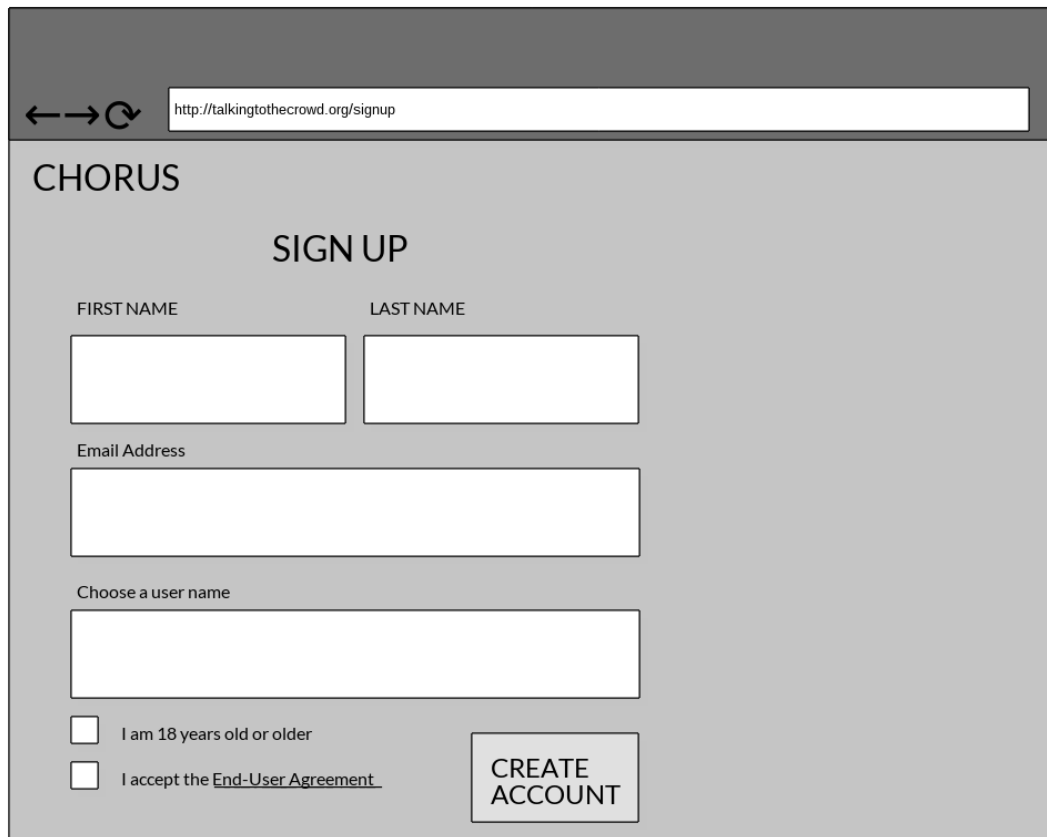
by clicking the three dots symbol on the rightmost section of the subsection. They can view the past conversations they had with Chorus by clicking to the subsection. Also that button enables end user to start a conversation with the Chorus system.

The end users will maintain their conversation with Chorus through a sub window on the right (chat window) which only pops out when user starts a conversation with Chorus. End Users can send photos to the Chorus via clicking the button at the bottom right corner of the chat window, they can send emojis through bottom left corner of the chat window. End Users are also able to draw pictures and forward them through clicking pencil button near photo button.

End Users can see the status of the Chorus from the top left corner of the chat window and they can close the window by clicking the cross button at the top right corner of the chat window. There will be a active Chorus logo (status indicator) which will symbolize the “typing” information about Chorus when a response is being composed by the workers.

End Users can view information page about Chorus by clicking to “About” button at top right corner, near “Sign out” button and they can see the Frequently Asked Questions page by clicking to the “FAQ” button near “About” button. End Users can report technical issues about Google system by clicking to the “Report” button from bottom left corner of the main page and they can report technical issues about Chorus system by clicking to the “Report” button from the top right corner of the chat window. End Users can sign out through clicking to the “Sign out” button near “About” button.

Some of other UI features for end user interface are represented below:



A screenshot of a web browser showing the sign-up page for 'CHORUS'. The browser's address bar displays 'http://talkingtothecrowd.org/signup'. The page has a grey background. At the top left, the word 'CHORUS' is written in a bold, sans-serif font. Below it, the title 'SIGN UP' is centered in a larger, bold, sans-serif font. The form consists of several input fields: two for 'FIRST NAME' and 'LAST NAME' side-by-side, a larger one for 'Email Address', and another for 'Choose a user name'. Below the 'Email Address' field, there are two checkboxes. The first checkbox is followed by the text 'I am 18 years old or older'. The second checkbox is followed by the text 'I accept the [End-User Agreement](#)'. To the right of these checkboxes is a rectangular button with the text 'CREATE ACCOUNT' in all caps.

←→↻ http://talkingtothecrowd.org/signup

CHORUS

SIGN UP

FIRST NAME LAST NAME

Email Address

Choose a user name

☐ I am 18 years old or older

☐ I accept the [End-User Agreement](#)

CREATE ACCOUNT

Figure 3: Sign Up Screen For End User



Sign in
with your Google Account

Email or phone

[Forgot email?](#)

[More options](#)

NEXT

Figure 4: Sign In Screen For End User

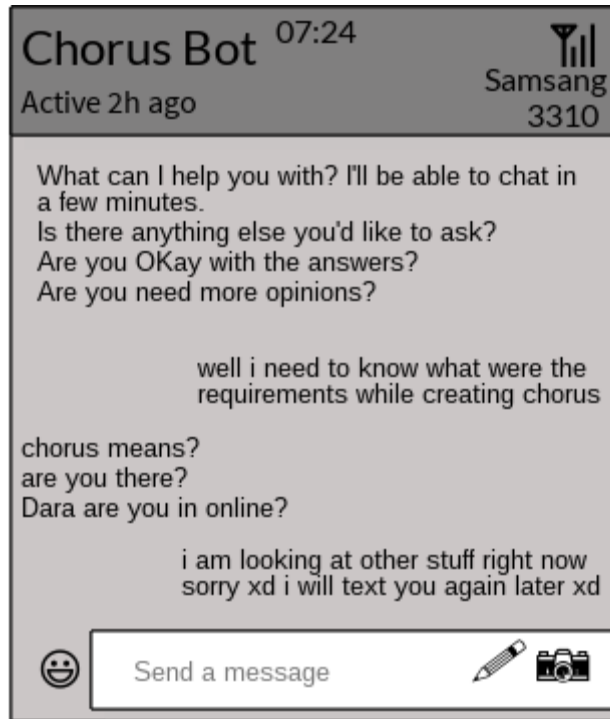


Figure 5: Android Main Screen For End User

Smartphone interface of the system is also handled by Google Hangouts application. End Users can interact with the system via main chat screen which has the same layout as the chat window in the browser. When the end user taps the message box, a default phone keyboard appears on the screen where the user can type. This keyboard is small enough to let the end user see chat and the message he/she is typing when it is on the screen. The end user can also send emojis by clicking the smiley face, drawings by clicking the pencil, photos by clicking the camera buttons. The end user can take new photos when he/she clicks the camera button.

For workers a specialized interface for Chorus is designed. On the main page there is three sub windows which shows the chat screen, shared chat history about end users and earnings box respectively. Workers are able to both propose responses and vote on the responses of others via main chat screen. They can also make use of a collective shared chat history (working memory) via shared chat history sub window on the right which allows them to maintain continuity throughout the conversation, even as some leave and new workers join. Above shared memory sub window there is an earnings window which shows workers how much money they earned from the system in total.

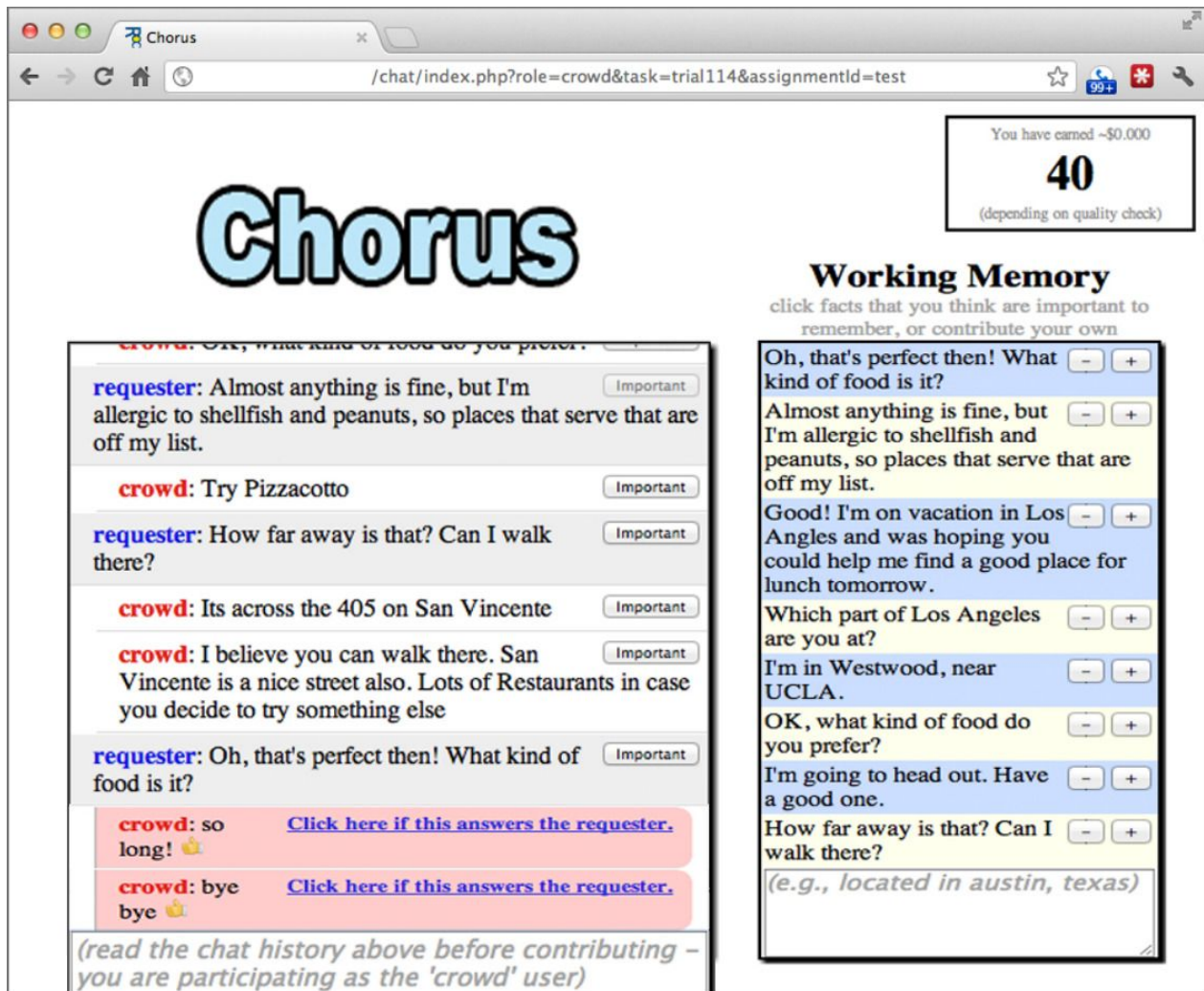


Figure 6: Main Screen For Worker

Another specialized interface is designed for the admin interface. On the main page there is a navigation bar which contains all the functionality tools that admin can utilize. On “Dashboard” page admin can view all the end user and worker activity with time informations and can view all warnings, errors, logs that come from the system. Other buttons on the bar are also available for utilizing different features. For example the “Bans” page shows ban protests, ban list, ban submissions and provides an environment for the management of banning mechanism of the system.

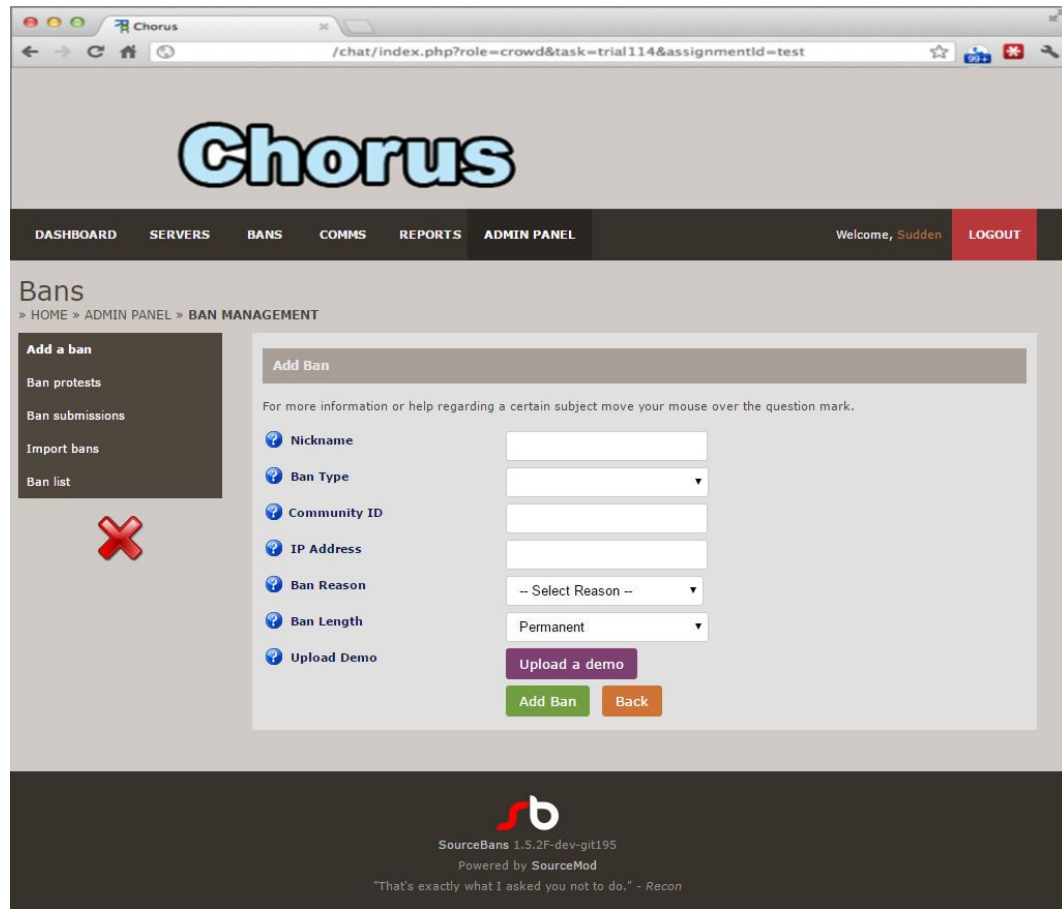


Figure 7: Main Screen For Admin

In the researcher interface there are a couple of features that researcher can utilize for the research, testing and development purposes. There is a “Manage Researches” page which redirects researcher to another page which has a set of functionalities that provide suitable and convenient environment for the research purposes. For instance, researchers can access databases to carry out their researches and they can maintain their tests and analyze results from this page. Also there is a “Results” page which shows the result history based on the past researches.

	Total Lines	Accurate Responses	Errors Made	Clarifications Asked	Questions Asked	Answers Provided	Memory Successes	Memory Failures
Memory #1	138	53	30	3	5	3	4	2
Memory #2	63	15	1	1	4	2	1	0
Memory #3	30	29	1	1	3	3	1	0
Memory #4	28	7	0	2	3	2	2	0

Figure 8: A Sample Data Result For Researcher

1.3.1.3. Hardware Interface

Hardware interface is mostly about servers and databases where the whole data about the past conversations, users and workers is stored and all the requests and data transmissions are handled in all internal modules. For server and database structure Amazon Web Services (AWS) is used. Server computers has following properties, 16GB RAM, 8 core processors, 160GB SSD Disk, 6TB Transfer.

1.3.1.4. Software Interface

- Bootstrap CSS Framework 4.0.0, JQuery 3.2 and Normalize CSS Framework is utilized to carry out front-end development.
- PHP 7.1, Laravel PHP Framework 5.4.22 is utilized to carry out back-end development.
- Database management is handled by SimpleDB.
- Engine for the Chorus system which handles the big data transmissions is implemented via Java.

1.3.1.5. Communication Interface

In Worker Module level a specialized and convenient remote network connection is provided by using a high number of routers and numerous switches. Workers use this network for communication between each other and accessing database to update or fetch some information about users when needed.

Apart from this specialized interface provided for workers, the system handles all the communication flow through conversation servers which maintains the network communication performance of the system. In both interfaces IEEE 802.1X protocol is used.

1.3.1.6. Memory Constraints

Memory constraints are not a big issue of Chorus system. RAM is used as the primary memory of the system. SSD Disk will be mainly used for data storage, therefore it is used as the secondary memory in the system. For management purposes of the database system a specialized hashing algorithm is used.

1.3.2. User Characteristics

There are three types of users in this project. One is end users. They are the most populous users. They are the users which ask questions and directly interact with workers. One must be at least 18 years old to be an end user. All they need to do is tick the “I am 18 years old or older” and sign up to system giving their full name and email address.

Another user type is worker. Their job is simple but important. They are responsible for answering the end user’s questions, voting for the good answers that other workers gave and saving the relevant information about the user. In case there are rogue workers who give faulty

or harmful information on purpose or chat in an explicit manner the other workers are responsible for reporting the rogue worker. Workers can be from any kind of profession however one must meet the requirements to become a worker, which are: understanding and speaking english fluently, being at least 18 years old, having a bank account, agree that they are not to say any harmful and/or explicit words.

Another type of user is researcher. They are responsible for handling the research. Researchers must have a PhD in Computer Science or similar programs. Researchers can not work as workers but they can moderate and ban workers. Researchers can also test and add new features.

Last type of user is administrator (IT people). As one can guess these people handles all system-wise works. They are in charge of maintaining the databases, keeping the website and corresponding web services up-to-date, keeping the software tools stable and compatible for users and workers, moderating and banning workers. Administrators must have PhD from Computer Science or similar programs.

1.3.3. Limitations

In Chorus system the main limitations are due to Network issues. Network-sourced limitations are mainly the cases where one of the users in the system fails to create and maintain any connections, such case happens when users or workers fail to create a connection between servers, database or their connection gets cut off while there is an ongoing conversation in system which they are occupying, then both parties won't be able to interact with the system.

In Chorus a special attention must be taken for the worker limitations since workers play an important role for the sustainability of the system. If worker number gets insufficient for handling ongoing conversations then some crucial requirements for the system may not be met. On the other hand the workers must be well-trained and they must not display any abuse of the system such as responding to user with malicious words etc. Same limitation is also present in users of the system, users also must not abuse the system, otherwise both of the cases may lead corruption in the system structure and layout. In cases where the administrators are not online in system for long periods of times the management, control mechanisms and maintenance of system may fail which may limit the system usage.

Safety and security considerations in the system is another area that requires special attention. These considerations lead database access to be strict, since all the private information noted about users, the important information about staff and the system is kept in databases. Therefore no direct access to database systems is possible unless the user is researcher or administrator.

There are no specific limitations due to server and database performance since these hardware components are designed and chosen with performance considerations beforehand.

Also there are no specific limitations due to user interfaces, account management and authorization processes since they are handled by Google and Google Hangouts.

1.4. Definitions

- AWS: Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
- PHP: It is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. PHP means PHP: Hypertext Preprocessor.
- SimpleDB: Amazon Simple Database Service (SimpleDB) is a highly available and flexible non-relational database that allows developers to request and store data, with minimal database management and administrative responsibility.
- HTTPS: It is a protocol for secure communication over a computer network which is widely used on the Internet. HTTPS consists of communication over Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security, or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.
- SHA512: SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the National Security Agency (NSA). Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity.
- JQuery: It is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.
- Laravel: It is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern.

2. References

This document is written with respect to the specifications of the document below:

29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering. (2013, January 24). Retrieved March 15, 2016, from <https://standards.ieee.org/findstds/standard/29148-2011.html>

Other sources:

Singh, M. P. (2005). *The practical handbook of Internet computing*. Boca Raton: Chapman & Hall/CRC.

Sommerville, I. (2016). *Software engineering*. Boston: Pearson Education Limited.

3. Specific Requirements

3.1. External Interfaces

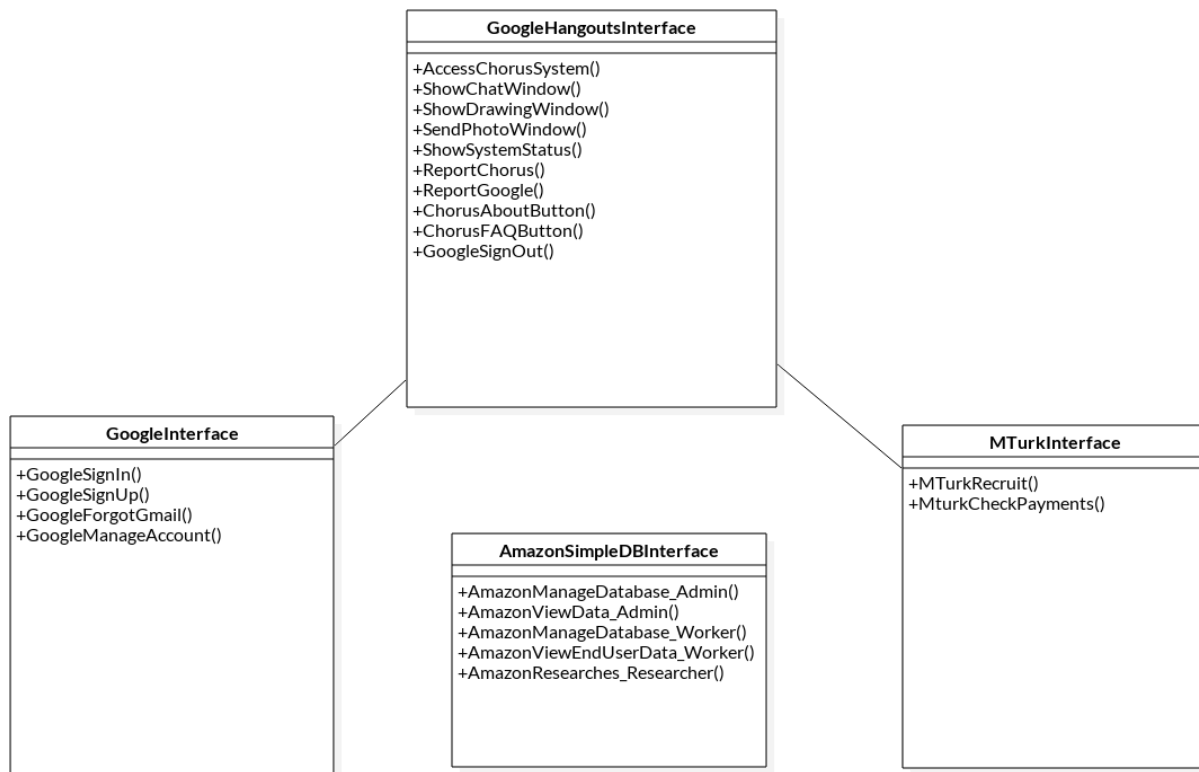


Figure 9: External Interfaces Class Diagram

As external interfaces we have Google for the authorization and account management processes, Google Hangouts for the end user interface and Amazon SimpleDB for building and maintaining database systems and frameworks for the development and MTurk for recruiting workers and handling their payments.

3.1.1. Google Hangouts Interface

req01: The interface will provide a contact window which will be used to handle end users' access to Chorus system.

req02: The interface will provide a chat window which will show end users chat history and enable them to send messages to the system.

req03: The interface will provide a drawing button to the end users in chat window which will open another sub window when clicked. This sub window will provide end users a drawing interface, to enable them to send drawings to the system.

req04: The interface will provide a button to the end users in chat window for uploading photos to the system which will open another sub window when clicked. This sub window will be used to provide end users a photo browsing interface, to enable them to send photos to the system.

req05: The interface will show the status of the system to the end users in the chat window.

req06: The interface will provide a reporting tool to the end users in chat window which will redirect end users to Chorus report page when clicked.

req07: The interface will provide a reporting tool to the end users in contact window which will redirect end users to Google report page when clicked.

req08: The interface will provide an about button to the end users which will enable end users to view information about the Chorus system.

req09: The interface will provide an FAQ button to the end users which will enable end users to view frequently asked questions about the Chorus system.

req10: The interface will provide an sign out button to the end users which will enable end users to sign out from the Google system.

3.1.2. Google Interface

req01: The interface will provide a sign in page to Google system for the end users.

req02: The interface will provide a sign up button which redirects end users to the sign up page of the Google system when clicked.

req03: The interface will provide a "Forgot email?" button which redirects end users to the save account page of the Google system when clicked.

req04: The interface will provide a "Manage account" button which redirects end users to the manage account page of the Google system when clicked.

3.1.3. MTurk Interface

req01: The interface will provide a “Recruit workers” page to the admins for recruiting workers to the Chorus system.

req02: The interface will provide a “Check payments” page to the admins for checking the payments of workers in the Chorus system.

3.1.4. Amazon SimpleDB Interface

req01: The interface will provide a “Manage database” page for admins to enable them to edit the content of the database or edit the settings of frameworks for the development.

req02: The interface will provide a “View data” page for admins to enable them to view all the information about system, researches, workers, researchers and end users.

req03: The interface will provide a “Manage database” page for workers to enable them to edit the content related with end users in the database.

req04: The interface will provide a “View end user data” page for workers to enable them to view all the relevant information saved about end users.

req05: The interface will provide a “Researches” page for researchers to handle their research needs.

3.2. Functions

3.2.1. End User

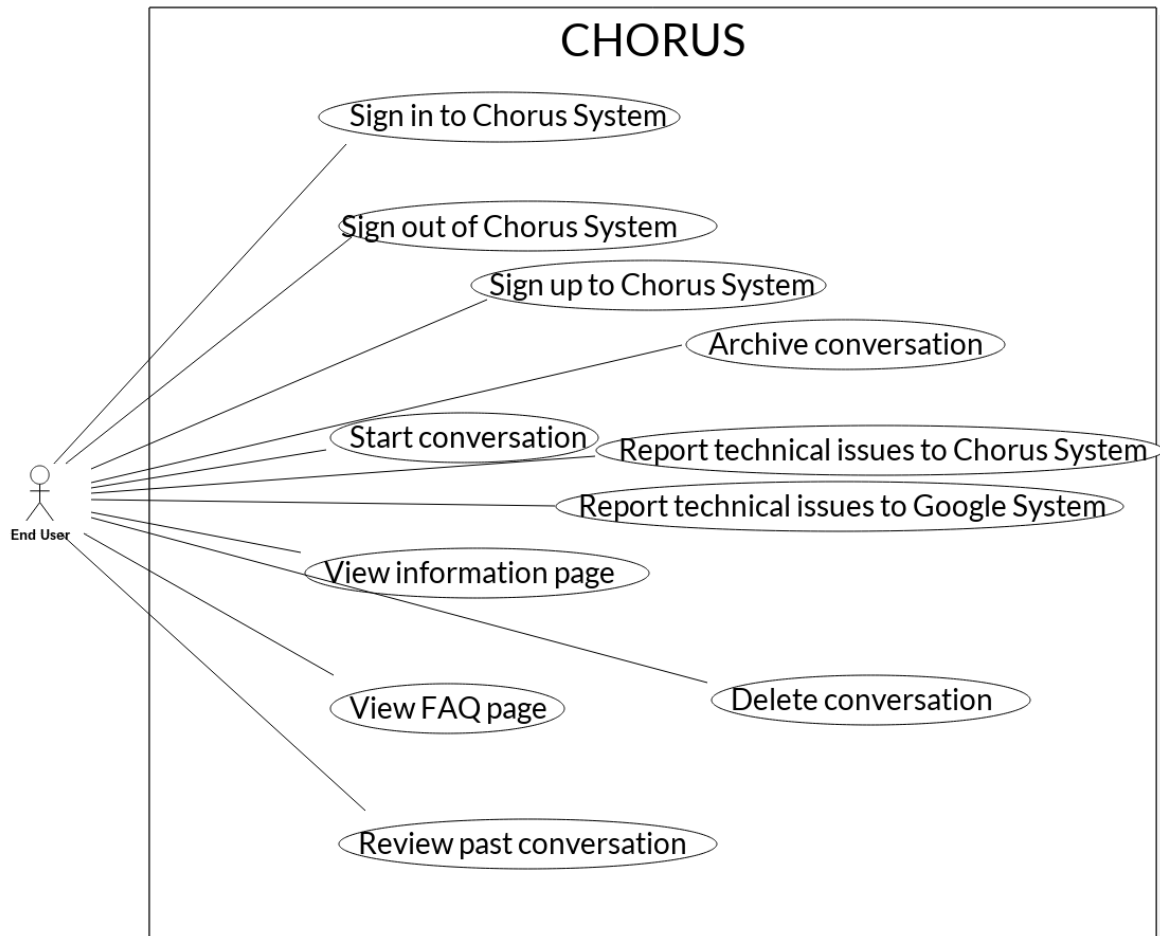


Figure 10: End User Use Case Model

Use Case ID	1
Use Case Name	Sign in
Actors	Chorus End User
Description	End User signs in to Chorus system in order to create conversation with workers.
Preconditions	End User exists in the system with a valid username and password.

Postconditions	End User is successfully signed in to Google system.
Normal Flow	<ol style="list-style-type: none"> 1. End User requests to enter Google system. 2. System prompts end user to sign in. 3. End User enters his/her own gmail and password. 4. System verifies entered information. 5. Google validates the End User. 6. System signs End User in to Google system.
Exceptions	<ol style="list-style-type: none"> 4. a. If the gmail and/or password entered by the End User is/are not valid: <ol style="list-style-type: none"> I. Google system displays “Invalid mail address or password” error message. II. Use case returns on step 2 of normal flow.

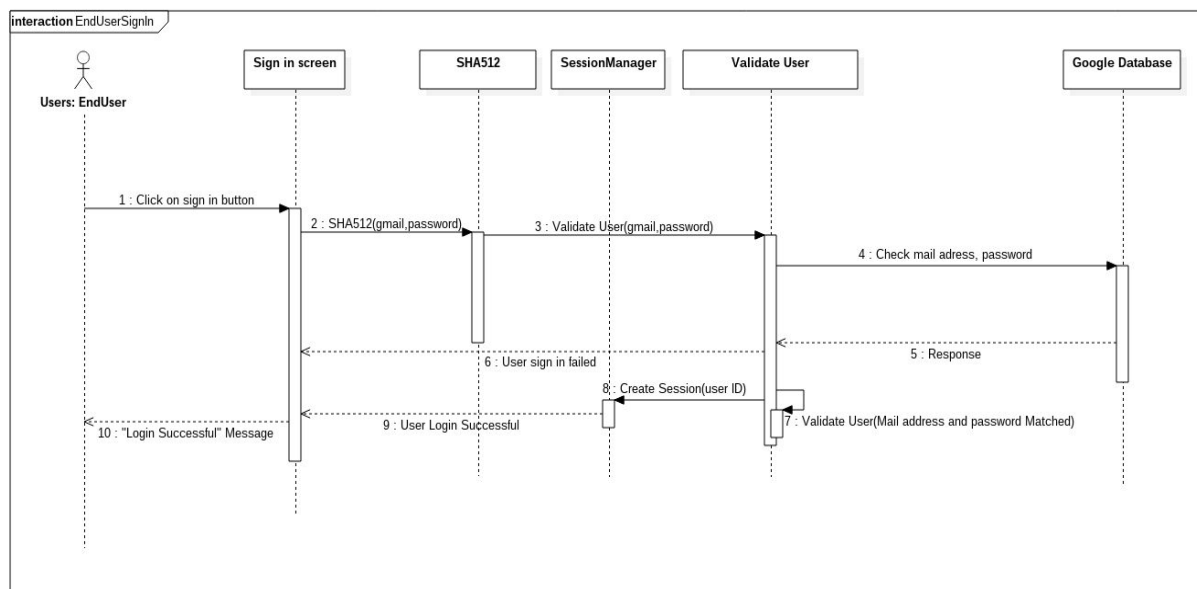


Figure 11: End User Sign In Sequence Diagram

Use Case ID	2
Use Case Name	Sign up
Actors	Chorus End User

Description	End User signs up to Chorus system in order to be able to access chat functionalities of the system.
Preconditions	End User that does not exist in the system fills in the blanks with valid first and last name, gmail, username and password.
Postconditions	End User is successfully signed up to Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. End User opens sign up page located at http://talkingtothecrowd.org/signup. 2. End User enters their first and last name, gmail, username and password. End User also ticks the “i am 18 years old or older” and “i accept the End-User Agreement” boxes 3. System verifies entered information. 4. Chorus validates the End User. 5. System successfully signs up the End User into Chorus system.
Exceptions	<p>3. a. If the first or last name, email, username or password entered by the observer is/are not valid:</p> <ol style="list-style-type: none"> I. Chorus system displays “ first or last name, email, username or password is not valid” error message. End User needs to enter valid format information. For example, for email, End User needs to enter valid email type. II. Use case returns on step 2 of normal flow. <p>3. b. If the email entered by the user exist in the system:</p> <ol style="list-style-type: none"> I. Chorus system displays “The email address you entered already exists in the system. Please enter another email address or click “forgot my password” button” error message. II. Use case returns on step 2 of normal flow <p>3. c. If the username entered by the End User</p>

	<p>exist in the system:</p> <ol style="list-style-type: none"> I. Chorus system displays “The username you entered already exists in the system. Please enter another username” error message. II. Use case returns on step 2 of normal flow
--	--

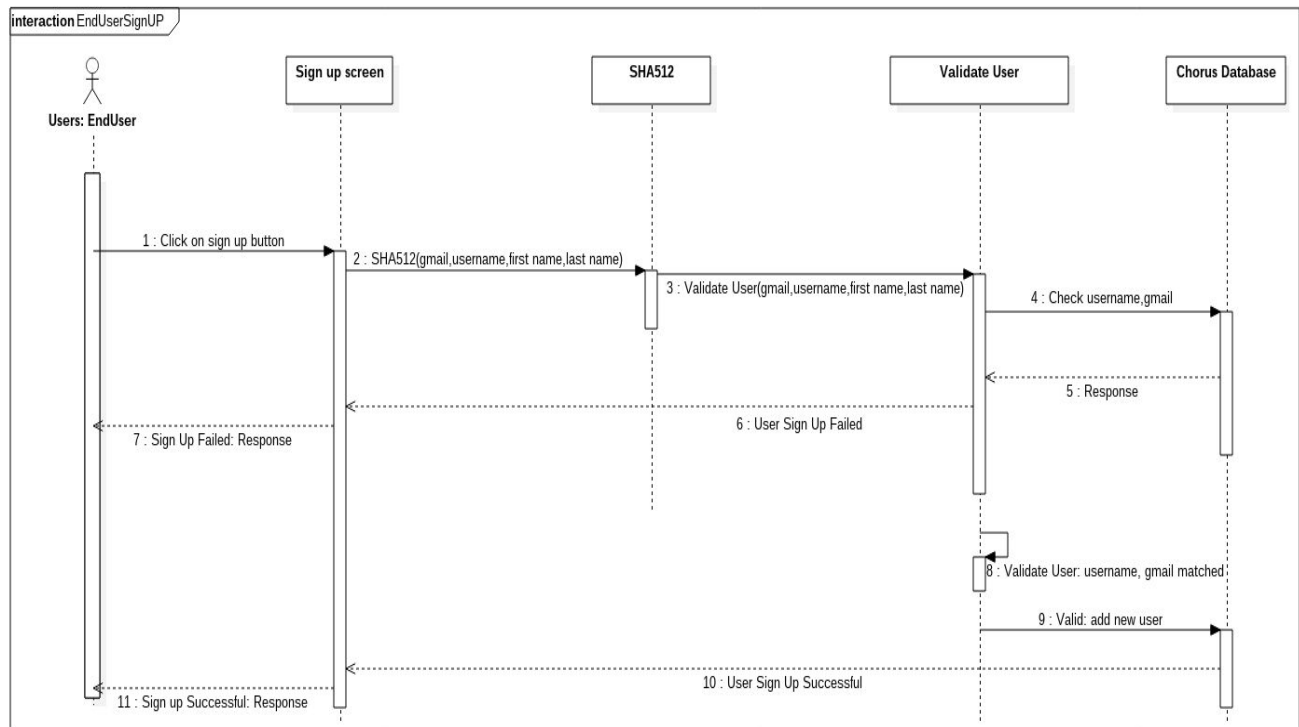


Figure 12: End User Sign Up Sequence Diagram

Use Case ID	3
Use Case Name	Sign Out
Actors	Chorus End User
Description	End User signs out of Google system in a successfully way.
Preconditions	End User must be signed in to Google system
Postconditions	End User is successfully signed out of Google system.

Normal Flow	<ol style="list-style-type: none"> 1. End User requests to sign out Google system. 2. System checks session information. 3. System signs out the End User from the Google system. 4. System returns End User to sign in page.
Exceptions	<ol style="list-style-type: none"> 2. a. If the session is not created by the system: <ol style="list-style-type: none"> I. Google system automatically goes to step 4 of the use case. 2. b. If the time of the session is exceeded: <ol style="list-style-type: none"> I. Google system automatically goes to step 4 of the use case.

Use Case ID	4
Use Case Name	Start conversation with Chorus
Actors	Chorus End User
Description	End User starts talking with the Chorus.
Preconditions	<p>End User must be signed in to Google System.</p> <p>End User must be signed up to Chorus System.</p>
Postconditions	End User successfully conversed with Chorus.
Normal Flow	<ol style="list-style-type: none"> 1. End User clicks the Chorus Bot's name on the left of the screen where conversations are. 2. A chat box pops up on the right. 3. User starts typing his/her message, emoji and/or sends a photo or a drawing. 4. Chorus responds. 5. Use case returns on step 3 of normal flow.
Exceptions	<ol style="list-style-type: none"> 4. a. If there is no workers to respond: <ol style="list-style-type: none"> I. A worker will be assigned to the user. II. Use case will return to step 4 of

	<p>normal flow.</p> <p>3.a. If user stops chatting:</p> <ol style="list-style-type: none"> I. Workers will leave the channel.
--	--

Use Case ID	5
Use Case Name	View information page
Actors	Chorus End User
Description	End User opens “information page” in order to get information about Chorus System.
Preconditions	End User must be in the Google Hangouts page.
Postconditions	End User is in “Information page”.
Normal Flow	<ol style="list-style-type: none"> 1. End User presses the “About” button. 2. System redirects End User to the “Information page”.
Exceptions	-

Use Case ID	6
Use Case Name	View past conversation
Actors	Chorus End User
Description	End User views past conversations in order to read them.
Preconditions	<p>End User is signed in to Google System.</p> <p>End User is signed up to Chorus System.</p> <p>End User has conversed with Chorus before.</p>
Postconditions	End User successfully reads past conversations.
Normal Flow	<ol style="list-style-type: none"> 1. End User opens the chat box of the Chorus Bot. 2. End User scrolls through the chat until he/she reads anything he/she wants.

	3. End User successfully read the past conversation.
Exceptions	2. a. If the past chat with Chorus bot is partially or fully deleted: <ul style="list-style-type: none"> I. End User can not read through those texts. II. System returns to step 1 of normal flow.

Use Case ID	7
Use Case Name	Archive conversation
Actors	Chorus End User
Description	End User archives conversation to hide it from the inbox until the next time he/she receives a message from Chorus.
Preconditions	End User has to be signed in to Google. End User must have conversed with Chorus. The conversation must not be archived already.
Postconditions	The conversation is successfully archived.
Normal Flow	<ol style="list-style-type: none"> 1. End User presses the 3 vertical dots next to the name of Chorus Bot in the contacts window. 2. End User presses archive button in the pop-up window. 3. Message is archived successfully.
Exceptions	-

Use Case ID	8
Use Case Name	Delete conversation
Actors	Chorus End User
Description	End User deletes past conversation from the

	contact window
Preconditions	End User must be signed in to Google. End User must have had a conversation with Chorus that is not deleted.
Postconditions	End User successfully deletes past conversation.
Normal Flow	<ol style="list-style-type: none"> 1. End User presses the 3 vertical dots next to the contacts name. 2. End User selects the delete option from the pop-up window. 3. System removes the contact from contact window.
Exceptions	<ol style="list-style-type: none"> 3. a. If the conversation can not be found: <ol style="list-style-type: none"> I. Chorus system displays “Conversation could not be found” error message. II. Use case returns on step 1 of normal flow.

Use Case ID	9
Use Case Name	Report technical issues to Google
Actors	Chorus End User
Description	End User reports technical issue to Google for the issue to be solved by Google.
Preconditions	End User must be signed in to Google.
Postconditions	The report is successfully sent to Google.
Normal Flow	<ol style="list-style-type: none"> 1. End User presses the report button at the bottom of contacts window. 2. End User fills the necessary boxes of the pop-up window. 3. End User presses “send” button to send the report.
Exceptions	-

Use Case ID	10
Use Case Name	Report technical issues to Chorus
Actors	Chorus End User
Description	End User reports technical issue to Chorus for the issue to be solved by Chorus.
Preconditions	End User is signed in to Google. End User is signed up to Chorus System. End User has conversed with Chorus.
Postconditions	The report is successfully sent to Chorus.
Normal Flow	<ol style="list-style-type: none"> 1. End User presses the report button at the bottom of “X” button of the chat window. 2. End User fills the necessary boxes of the pop-up window. 3. End User presses “send” button to send the report.
Exceptions	-

3.2.2. Workers

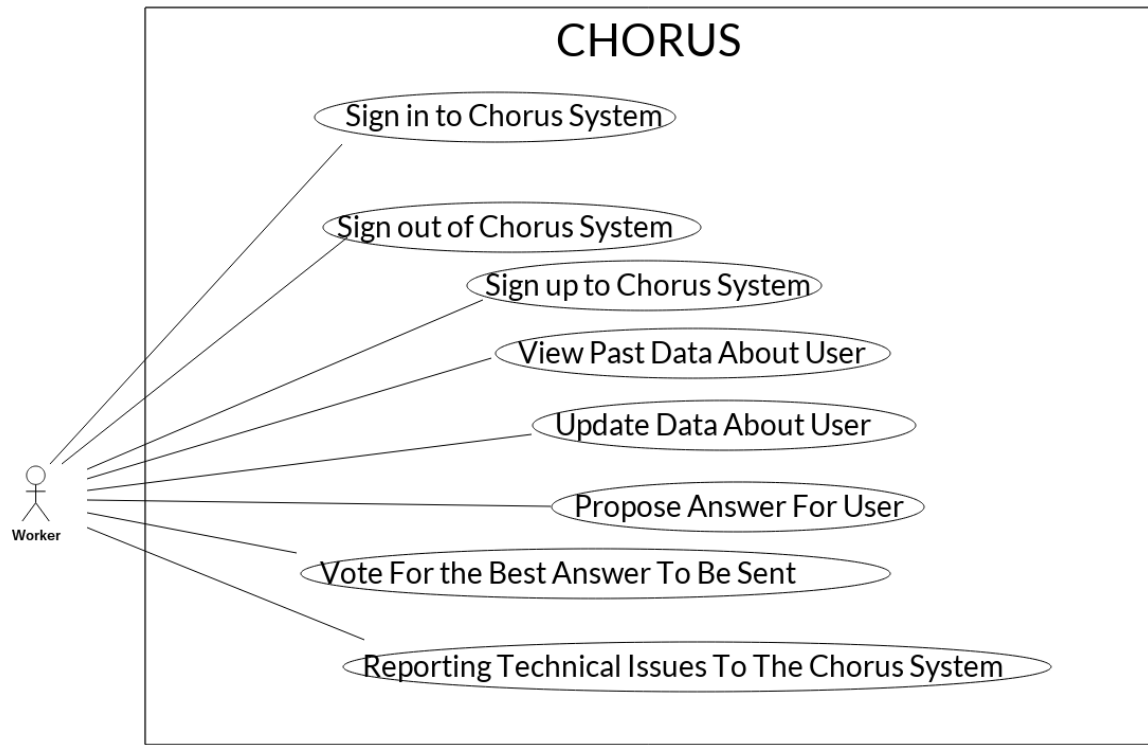


Figure 13: Worker Use Case Model

Use Case ID	1
Use Case Name	Sign in
Actors	Chorus Worker
Description	Worker signs in to the Chorus System successfully.
Preconditions	Worker is not already signed in to Chorus System. Worker is signed up to Chorus System. Worker is not banned from the Chorus System.
Postconditions	Worker is successfully signed in to Chorus System.
Normal Flow	1. Worker opens the worker sign in page.

	<ol style="list-style-type: none"> 2. Worker enters his/her username and password. 3. Chorus System checks the information. 4. Chorus System validates the worker. 5. Worker is signed in to Chorus System.
Exceptions	<ol style="list-style-type: none"> 3. a. If the username and/or password entered by the worker is/are not valid: <ol style="list-style-type: none"> III. Chorus system displays “Invalid username or password” error message. IV. Use case returns on step 2 of normal flow.

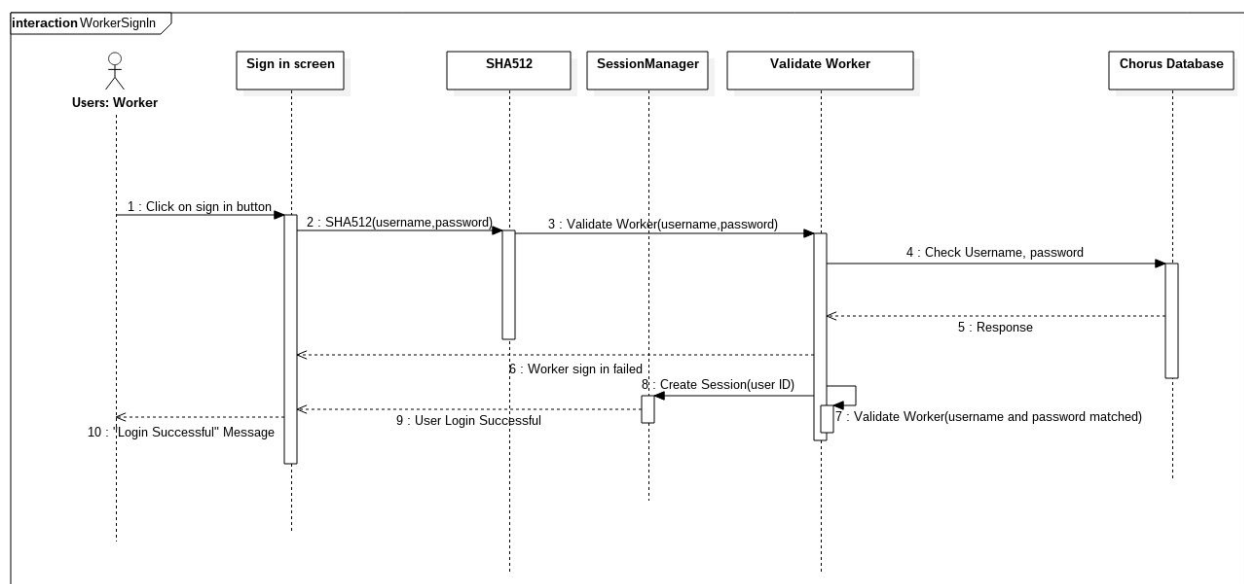


Figure 14: Worker Sign In Sequence Diagram

Use Case ID	2
Use Case Name	Sign Up
Actors	Chorus Worker
Description	Worker signs up to Chorus System in order to be able to send and receive messages.

Preconditions	<p>Worker chooses a valid username.</p> <p>Worker chooses a valid password.</p> <p>Worker is not already signed up.</p> <p>Worker meets the minimum requirements specified on MTurk.</p>
Postconditions	Worker is successfully signed up for Chorus.
Normal Flow	<ol style="list-style-type: none"> 1. Worker presses the join button on MTurk. 2. MTurk redirects the worker to the worker signup page. 3. Worker types a username and a password. 4. Chorus System checks the information. 5. Chorus System validates the worker. 6. Chorus System adds the worker to the database. 7. User is given a link to the worker interface where he/she can do his/her job after signing in.
Exceptions	<ol style="list-style-type: none"> 4. a. If the username and/or password entered by the worker is/are not valid: <ol style="list-style-type: none"> I. Chorus system displays “Invalid username or password” error message. II. Use case returns on step 3 of normal flow.

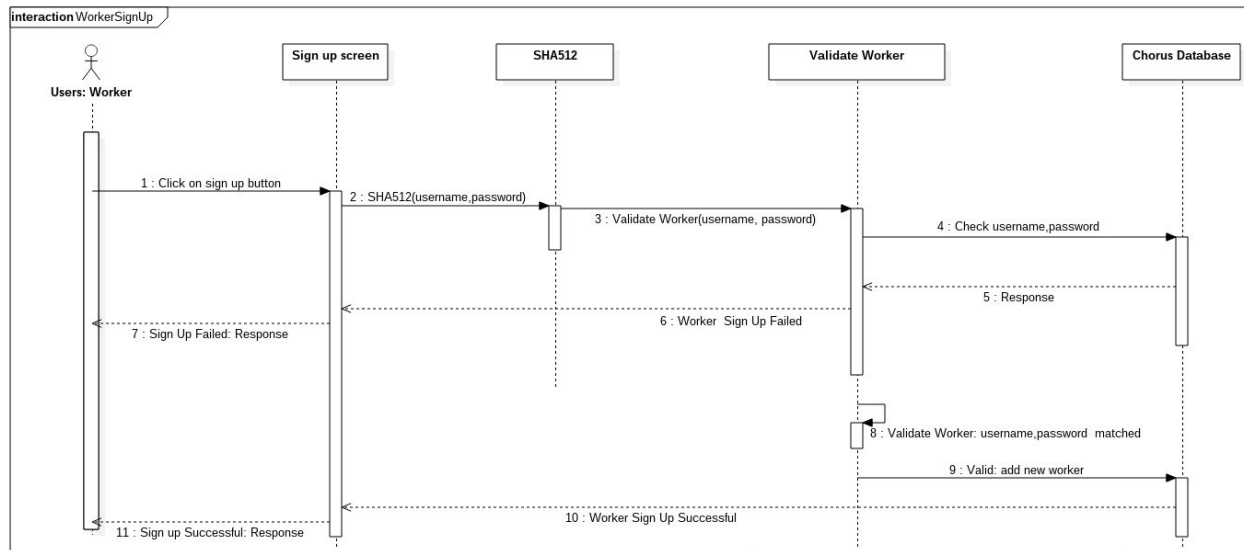


Figure 15: Worker Sign Up Sequence Diagram

Use Case ID	3
Use Case Name	Sign Out
Actors	Chorus Worker
Description	Worker signs out of the chorus system in a successful way.
Preconditions	Worker must be signed in to chorus system.
Postconditions	Worker is successfully signed out of the Chorus System.
Normal Flow	<ol style="list-style-type: none"> 1. Worker requests to sign out of the Chorus System. 2. System checks session information. 3. System signs out the Worker from the Chorus System. 4. System returns Worker to sign in page.
Exceptions	<ol style="list-style-type: none"> 2. a. If the session not created by the system: <ol style="list-style-type: none"> 1. Chorus system automatically goes to step 4 of normal flow. 2. b. If the time of the session is exceeded: <ol style="list-style-type: none"> 1. Chorus system automatically goes to step 4 of normal flow.

Use Case ID	4
Use Case Name	View past relevant data about end user
Actors	Chorus Worker
Description	Worker looks at the past data about the user in order to be more effective at answering End User's questions.
Preconditions	Worker must be signed in to Chorus System. End User must have past data in the Chorus System.
Postconditions	Worker successfully gets relevant data about End User.
Normal Flow	<ol style="list-style-type: none"> 1. Worker joins the chat room. 2. Worker scrolls through the working memory to read the past data about the End User. 3. Worker successfully gets the data about the End User.
Exceptions	-

Use Case ID	5
Use Case Name	Update data about user
Actors	Chorus Worker
Description	Worker updates the data about the user in order to contribute to system accuracy.
Preconditions	Worker must be signed in to Chorus System. The data must receive enough votes from the workers.
Postconditions	The data is successfully updated to the Chorus System.
Normal Flow	<ol style="list-style-type: none"> 1. Worker types important data about the End User to the "Working Memory" section in his/her interface.

	<ol style="list-style-type: none"> 2. Workers upvote the data. 3. Chorus System checks the minimum number of votes required and total number of votes the data has. 4. The data gets pushed to the Chorus System.
Exceptions	<ol style="list-style-type: none"> 3. a. If the data does not get enough votes after 3 minutes or after 8 other entries have been added to the “Working Memory” section: <ol style="list-style-type: none"> I. Chorus System removes the data from “Working Memory” section. II. Use case returns on step 1 of normal flow. 3. b. If the data gets 2 downvotes: <ol style="list-style-type: none"> I. Chorus System removes the data from “Working Memory” section. II. Chorus System sends a report to the admins in order for them to review and penalize the worker. III. Use case returns on step 1 of normal flow. 3. c. If there is less than 4 workers in the chat room: <ol style="list-style-type: none"> I. The data will be removed with only 1 downvote. II. The data will be updated with minimum (# of workers-1) upvotes. III. Use case returns on step 1 of normal flow

Use Case ID	6
Use Case Name	Propose answer for End User.
Actors	Chorus Worker
Description	Worker types an answer in order to respond to End User.
Preconditions	Worker must be signed in to Chorus System. The answer must receive enough votes from the workers.
Postconditions	The answer is successfully given to End User.

Normal Flow	<ol style="list-style-type: none"> 1. Worker types his/her response in the chat box under the “Chorus” sign. 2. Workers upvote the respond. 3. Chorus System checks the minimum number of votes required and total number of votes the response has. 4. The response is successfully given to End User.
Exceptions	<ol style="list-style-type: none"> 3. a. If the answer does not get enough votes after 3 minutes or after 8 other entries have been added to the chat section: <ol style="list-style-type: none"> I. Chorus System removes the answer from chat section. II. Use case returns on step 1 of normal flow. 3. b. If the there is less than 4 workers in the chat: <ol style="list-style-type: none"> I. The answer will be given with minimum (# of workers-2) upvotes. II. Use case returns on step 1 of normal flow.

Use Case ID	7
Use Case Name	Vote for answer to be sent to user
Actors	Chorus Worker
Description	Worker votes for responses to send them to users.
Preconditions	Worker is signed in to Chorus System. A potential answer is posted on chat box.
Postconditions	The response is successfully upvoted.
Normal Flow	<ol style="list-style-type: none"> I. Worker presses the “click here if this answer the user” button on the right of the answer. II. Chorus System logs the worker who gave the answer, the worker who upvoted and the relevant chat history, and sends them to the admin in order to review and reward the worker who gave the answer.

	III. The answer is successfully upvoted.
Exceptions	<p>1. a. If the answer is already upvoted by the worker:</p> <ul style="list-style-type: none"> I. Worker will be unable to upvote the same answer again. II. Use case returns on step 1 of the normal flow. <p>1. b. If the answer has already gained enough upvotes:</p> <ul style="list-style-type: none"> I. Worker will be unable to upvote the posted answer. II. Use case returns on step 1 of the normal flow.

Use Case ID	8
Use Case Name	Reporting technical issues to the Chorus System.
Actors	Chorus Worker
Description	Worker reports technical issues to the Chorus System for the system to perform better.
Preconditions	Worker must be signed up to Chorus System.
Postconditions	The report is successfully sent.
Normal Flow	<ol style="list-style-type: none"> 1. Worker finds a fault. 2. Worker clicks to the "Report" button in the worker interface. 3. Worker enters the necessary information. 4. The report is successfully sent.
Exceptions	-

3.2.3. Researcher

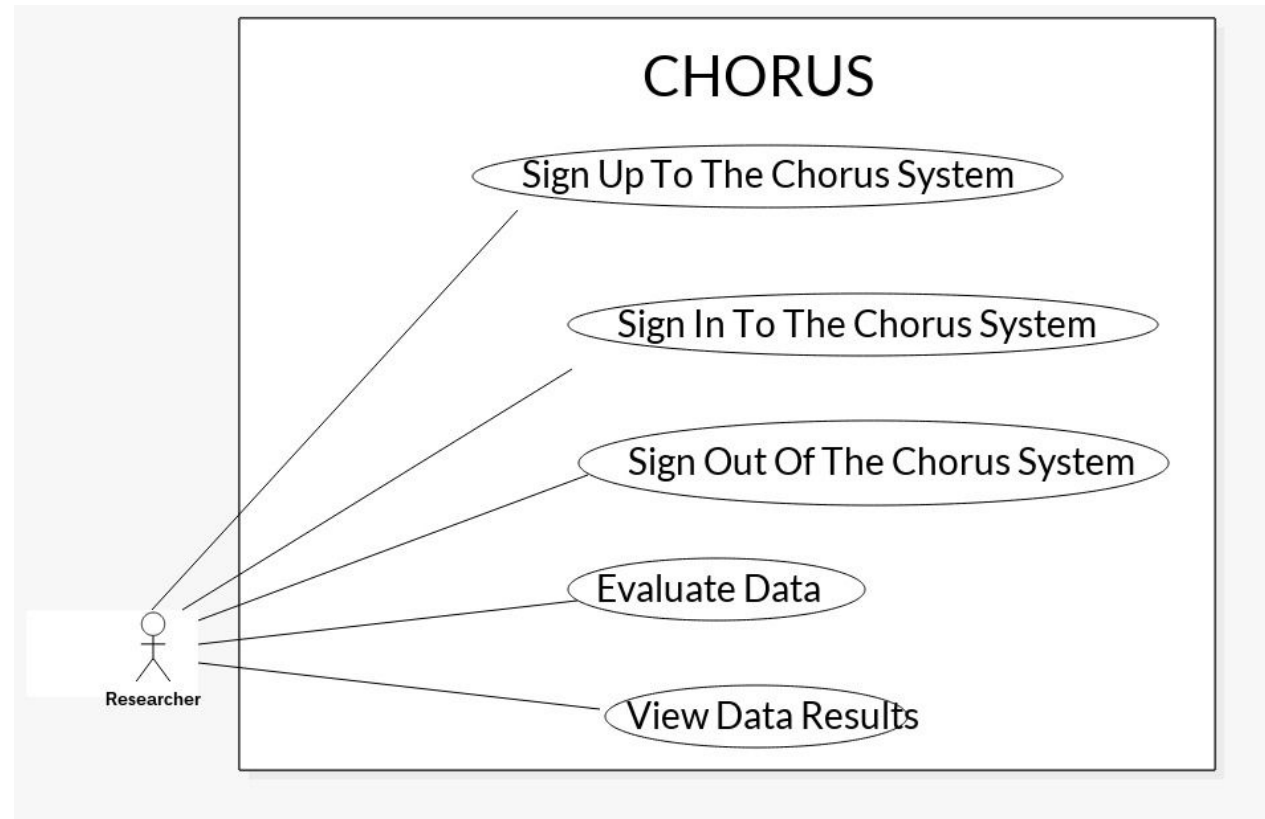


Figure 16: Researcher Use Case Model

Use Case ID	1
Use Case Name	Sign in
Actors	Chorus Researcher
Description	Researcher signs into Chorus system in order to analyze data that is collected from the conversations.
Preconditions	Researcher must be signed up to Chorus System. Researcher is not already signed in to Chorus System
Postconditions	Researcher is successfully signed in to Chorus system.
Normal Flow	1. Researcher opens the researcher sign

	<p>in page.</p> <ol style="list-style-type: none"> 2. Researcher enters his/her username and password. 3. Chorus System checks the information. 4. Chorus System validates the Researcher. 5. Researcher is signed in to Chorus System.
Exceptions	<ol style="list-style-type: none"> 3. a. If the username and/or password entered by the researcher is/are not valid: <ol style="list-style-type: none"> I. Chorus system displays “Invalid username or password” error message. II. Use case returns on step 2 of normal flow.

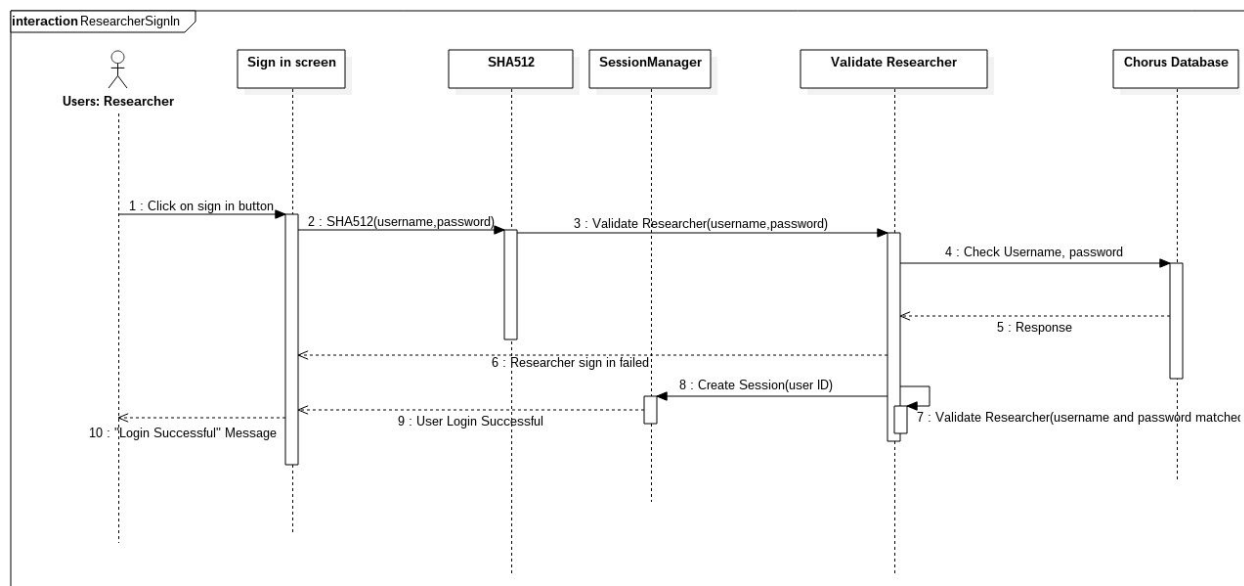


Figure 17: Researcher Sign in Sequence Diagram

Use Case ID	2
Use Case Name	Sign Up
Actors	Chorus Researcher
Description	Researcher signs up to Chorus System in

	order to be able to analyze the data collected from conversations.
Preconditions	<p>Researcher chooses a valid username.</p> <p>Researcher chooses a valid password.</p> <p>Researcher is not already signed up.</p> <p>Researcher is chosen beforehand.</p>
Postconditions	Researcher is successfully signed up for Chorus.
Normal Flow	<ol style="list-style-type: none"> 1. Researcher receives the one time use sign up link. 2. Researcher types a username and a password. Researcher also ticks "I accept the Researcher Agreement" box. 3. Chorus System checks the information. 4. Chorus System validates the Researcher. 5. Chorus System adds the researcher to the database. 6. Researcher is given a link to the researcher interface where he/she can do his/her job after signing in.
Exceptions	<ol style="list-style-type: none"> 3. a. If the username and/or password entered by the researcher is/are not valid: <ol style="list-style-type: none"> I. Chorus system displays "Invalid username or password" error message. II. Use case returns on step 2 of normal flow.

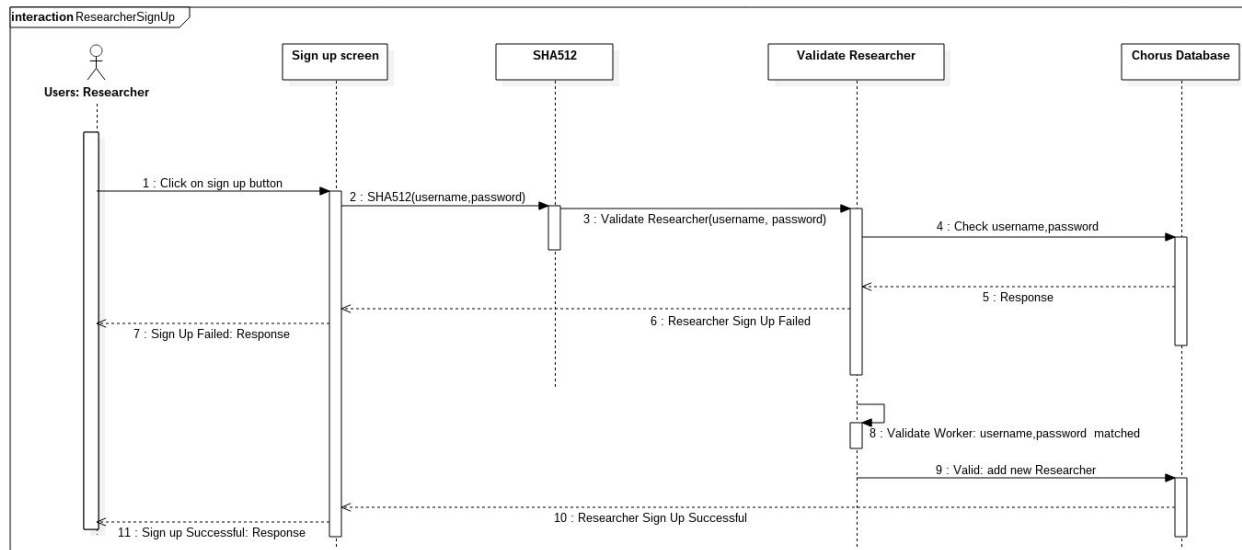


Figure 18: Researcher Sign up Sequence Diagram

Use Case ID	3
Use Case Name	Sign Out
Actors	Chorus Researcher
Description	Researcher signs out of the Chorus System in a successful way.
Preconditions	Researcher must be signed in to Chorus System.
Postconditions	Researcher is successfully signed out of the Chorus System.
Normal Flow	<ol style="list-style-type: none"> 1. Researcher requests to sign out of the Chorus System. 2. System Checks session information. 3. System signs out the researcher from the Chorus System. 4. System returns researcher to sign in page.
Exceptions	<ol style="list-style-type: none"> 2.a. If the session not created by the system: <ol style="list-style-type: none"> I. Chorus system automatically goes to step 4 of normal flow. 2. b. If the time of the session is exceeded: <ol style="list-style-type: none"> I. Chorus system automatically goes to

	step 4 of normal flow.
--	------------------------

Use Case ID	4
Use Case Name	View data results
Actors	Chorus Researcher
Description	Researcher views the data from the conversations in order to analyze it.
Preconditions	Researcher must be signed in to Chorus System. There must be data from the conversations in the database.
Postconditions	Researcher successfully views the data.
Normal Flow	<ol style="list-style-type: none"> 1. Researcher goes to the researcher database interface on Amazon SimpleDB. 2. Researcher presses the view data button. 3. Chorus System returns the data. 4. Researcher successfully views data.
Exceptions	<ol style="list-style-type: none"> 3.a. If the result can not be found by the system: <ol style="list-style-type: none"> I. Chorus system automatically goes to step 1 of normal flow.

Use Case ID	5
Use Case Name	Evaluate data
Actors	Chorus Researcher
Description	Researcher evaluates the viewed data in order to have a result.
Preconditions	Researcher must be signed in to Chorus System. There must be data from the conversations in the database.

	Researcher must have viewed new data.
Postconditions	Researcher successfully comes up with a result.
Normal Flow	<ol style="list-style-type: none"> 1. Researcher evaluates data. 2. Researcher gets a result. 3. Researcher pushes the result into the database.
Exceptions	

3.2.4. Administrator

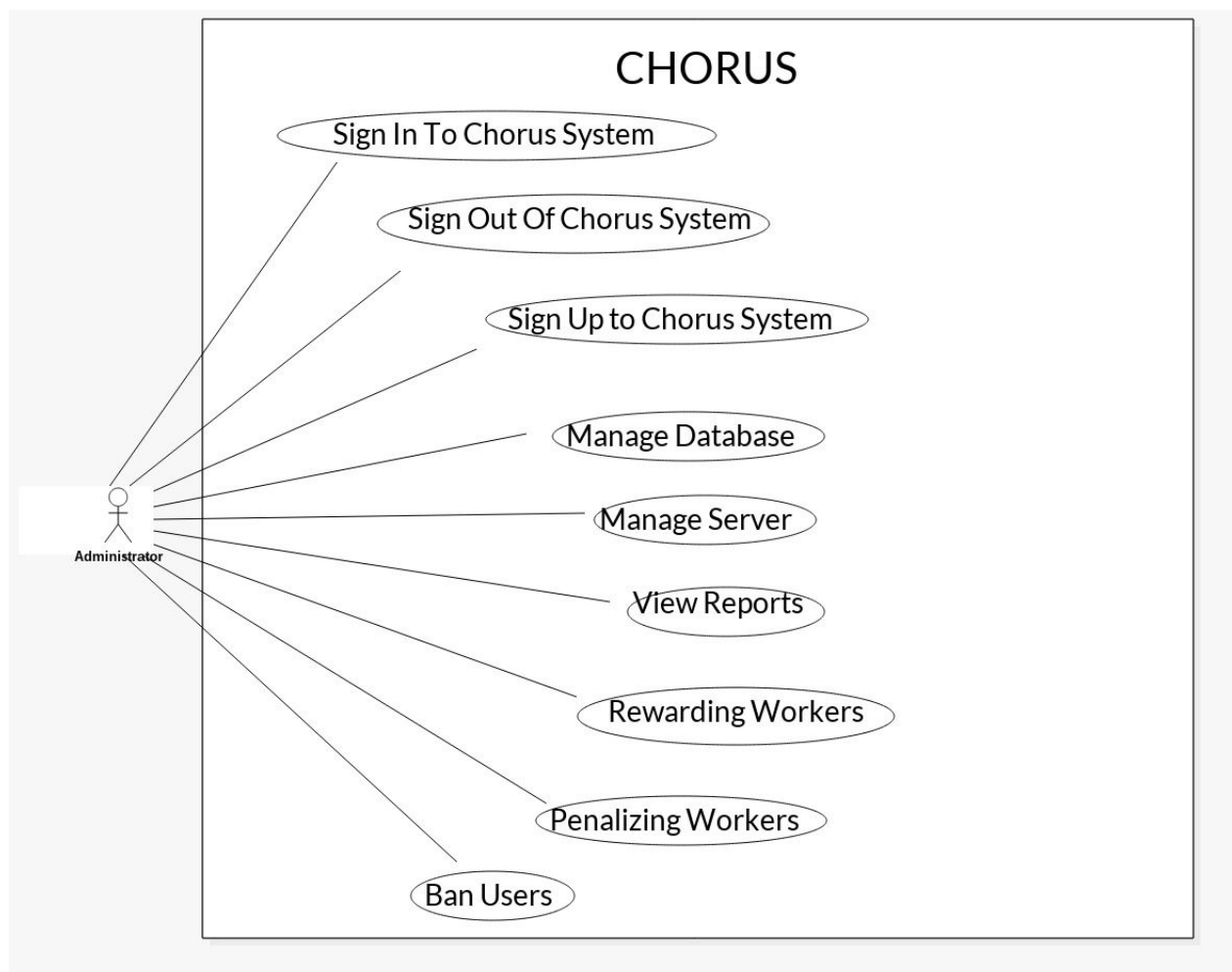


Figure 19: Administrator Use Case Model

Use Case ID	1
Use Case Name	Sign in
Actors	Chorus Administrator
Description	Administrator signs in to Chorus system in order to be able to do his/her job.
Preconditions	Administrator exists in the system with a valid username and password.
Postconditions	Administrator is successfully signed in to Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to enter Chorus system. 2. System prompts administrator to sign in. 3. Administrator enters his/her own username and password. 4. Chorus System checks the information. 5. Chorus System validates the administrator. 6. System logs administrator into Chorus system.
Exceptions	<ol style="list-style-type: none"> 4. a. If the username and/or password entered by the administrator is/are not valid: <ol style="list-style-type: none"> I. Chorus system displays “Invalid username or password” error message. II. Use case returns on step 3 of normal flow.

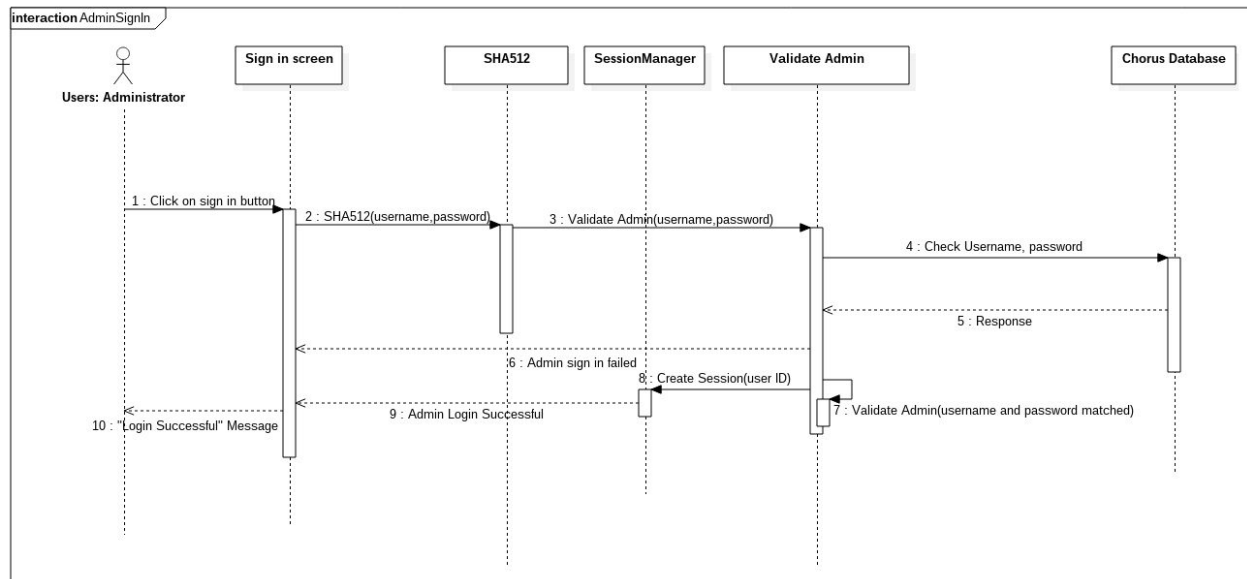


Figure 20: Administrator Sign In Sequence Diagram

Use Case ID	2
Use Case Name	Sign Up
Actors	Chorus Administrator
Description	Administrator signs up to Chorus system in order to be able to access the system management page of the system.
Preconditions	Administrator chooses a valid username. Administrator chooses a valid password. Administrator is not already signed up. Administrator will be chosen beforehand.
Postconditions	Administrator is successfully signed up to Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator opens the single use sign up page. 2. Administrator enters username and password. Administrator also ticks the "I accept the Administrator Agreement" box. 3. Chorus System checks the

	<p>information.</p> <ol style="list-style-type: none"> Chorus System validates the administrator. Chorus system adds the administrator to the database. Administrator is given a link which redirects administrator to admin interface when clicked.
Exceptions	<p>3.a If the username and/or password entered by the administrator is/are not valid:</p> <ol style="list-style-type: none"> Chorus system displays “Invalid username or password” error message. Use case returns on step 2 of normal flow.

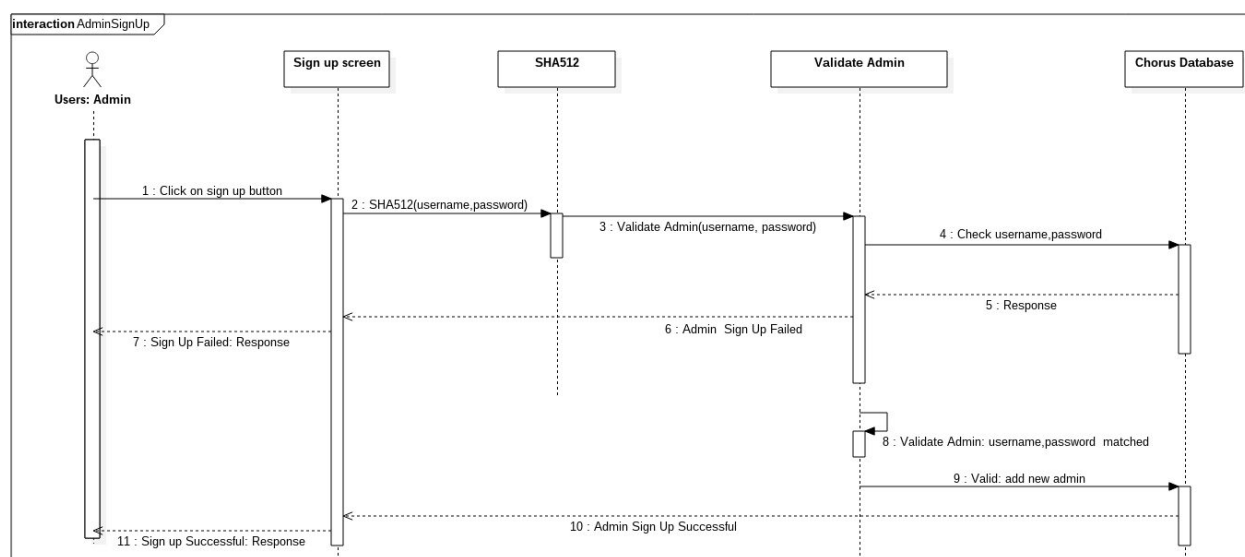


Figure 21: Administrator Sign Up Sequence Diagram

Use Case ID	3
Use Case Name	Sign Out
Actors	Chorus Administrator
Description	Administrator signs out from Chorus system in a successfully way.

Preconditions	Administrator must be already signed in to Chorus system.
Postconditions	Administrator is successfully signed out Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to sign out Chorus system. 2. System checks session information. 3. System logs out administrator from the Chorus System. 4. System returns to the administrator sign in page.
Exceptions	<ol style="list-style-type: none"> 2. a. If the session is not created by the system: <ol style="list-style-type: none"> I. Chorus system automatically goes to step 4 of the normal flow. 2. b. If the time of the session is exceeded: <ol style="list-style-type: none"> I. Chorus system automatically goes to step 4 of the normal flow.

Use Case ID	4
Use Case Name	Manage Database
Actors	Chorus Administrator
Description	Administrator manage databases. For example, s/he can easily add a new worker type in the "Manage Database" section.
Preconditions	Administrator must be signed in to Chorus system.
Postconditions	Administrator successfully manages the databases of Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to manage databases of the system by clicking "Manage Databases" button in "Admin Dashboard" 2. System loads "Manage Databases" page. 3. Administrator selects which database s/he wants to manage.

	<ol style="list-style-type: none"> 4. Administrator does operation in the selected database and save it. 5. System displays the result of the operation 6. System automatically returns “Admin Dashboard” page.
Exceptions	<ol style="list-style-type: none"> 4. a. If the changes do not save in the database: <ol style="list-style-type: none"> I. Chorus system displays “The changes were not saved in the database” error message. Administrator needs to do it again. II. Chorus system automatically goes to step 3 of the normal flow.

Use Case ID	5
Use Case Name	Manage Server
Actors	Chorus Administrator
Description	Administrator manages servers. For example, s/he can balance the conversations volume and flow via “Manage Server” section.
Preconditions	Administrator must be signed in to Chorus system.
Postconditions	Administrator successfully manages the servers of Chorus system.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to manage servers of the system by clicking “Manage Servers” button in “Admin Dashboard” 2. System loads “Manage Servers” page. 3. Administrator selects which operation s/he wants to do in server structure 4. Administrator does the operation and saves it. 5. System displays the result of the operation. 6. System automatically returns “Admin

	Dashboard” page.
Exceptions	<p>4. a. If the changes do not save in the database:</p> <ol style="list-style-type: none"> I. Chorus system displays “The changes were not saved in the database” error message. Administrator needs to do it again. II. Chorus system automatically goes to step 3 of the normal flow.

Use Case ID	6
Use Case Name	View Reports
Actors	Chorus Administrator
Description	Administrator views the reports that is sent from the workers and end users.
Preconditions	Administrator must be signed in to Chorus system.
Postconditions	Administrator successfully views the reports that is sent from workers and end users.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to view the reports by clicking “Reports” button in “Admin Dashboard” 2. System loads “Reports” page. 3. Administrator selects report s/he wants to view. 4. Administrator views the selected report.
Exceptions	<p>3. a. If there is no report to be seen:</p> <ol style="list-style-type: none"> I. System goes to step 2 of the normal flow.

Use Case ID	7
Use Case Name	Banning End Users

Actors	Chorus Administrator
Description	Administrator bans the end users for a time period or indefinitely.
Preconditions	Administrator must be signed in to Chorus system.
Postconditions	Administrator successfully bans the end user.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to ban end user by clicking "Bans" button in "Admin Dashboard" 2. System loads "Bans" page. 3. Administrator types the username of the end user s/he wants to ban to the search bar. 4. Administrator adjusts the time limit that end user will be prohibited from the system. 5. Administrator bans the end user. 6. System displays the result of the operation. 7. System automatically returns "Admin Dashboard" page.
Exceptions	<ol style="list-style-type: none"> 5. a. If the process is interrupted: <ol style="list-style-type: none"> I. Chorus system displays "The process is interrupted" error message. Administrator needs to do it again. II. Chorus system automatically goes to step 3 of the normal flow.

Use Case ID	8
Use Case Name	Penalize workers
Actors	Chorus Administrator
Description	Administrator penalizes the workers by lowering the payments of them or suspending them for a time period.
Preconditions	Administrator must be signed in to Chorus system.

Postconditions	Administrator successfully penalizes the worker.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to penalize workers by clicking "Bans" button in "Admin Dashboard" 2. System loads "Bans" page. 3. Administrator types the username of the worker s/he wants to penalize to the search bar. 4. Administrator selects the penalizing method from the list. 5. Administrator penalizes the worker. 6. System displays the result of the operation. 7. System automatically returns "Admin Dashboard" page.
Exceptions	<ol style="list-style-type: none"> 5. a. If the process is interrupted: <ol style="list-style-type: none"> I. Chorus system displays "The process is interrupted" error message. Administrator needs to do it again. II. Chorus system automatically goes to step 4 of the normal flow.

Use Case ID	9
Use Case Name	Rewarding Workers
Actors	Chorus Administrator
Description	Administrator rewards the workers in accordance with their performance.
Preconditions	Administrator must be signed in to Chorus system.
Postconditions	Administrator successfully rewards the worker.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator requests to reward worker by clicking "View Staff" button in "Admin Dashboard" 2. System loads "View Staff" page. 3. Administrator types the name of worker s/he wants to reward to the

	<p>search bar.</p> <ol style="list-style-type: none"> 4. Administrator selects the rewarding method from the list. 5. Administrator rewards the worker. 6. System displays the result of the operation. 7. System automatically returns to the "Admin Dashboard Page".
Exceptions	<ol style="list-style-type: none"> 5. a. If the worker is currently banned: <ol style="list-style-type: none"> I. System responds with "Worker is banned, can not reward worker" message. II. System returns to step 2 of normal flow

3.3. Usability Requirements

The interfaces will be functional and practical for all type of users. The system structure is designed as simple as possible. End users can easily use the system by only clicking to the "Chorus Bot" button on the contact window in the main screen of Google Hangouts. All the features that they can do are available with big font size and flashy designs in the screen, since end users can also be elder people. The most complex layout and structure in the systems are conversation servers and database. Their management system will only be accessible by admins who are authorized and educated people about these structures. For the other interfaces provided by database to workers and researchers the functions to be used will be simple and convenient, therefore there will not be any ambiguity for these users while using the system. For example there will be a search bar for the workers in the database interface designed for them, to ease their job while they are trying to find some specific end user's information. On the other hand a feedback system is implemented in Chorus system to ensure a smoother usage of the system by all users.

3.4. Performance requirements

Since the system maintenance mainly depends on network structures, the performance criteria relies on the servers and database. Servers will be designed to handle possible delays of pages (especially in cases hat high overload occuring) and implementations done with Php will work fast. Data transmissions will be smooth. Servers must handle at least 1000 users connected at the same time. In cases where the extra action is needed to balance the connection overload admins must be able to access the Php codes and edit them. Also in

front-end development the CSS frameworks are designed to be fast and efficient so there will not be any website-sourced problems (such as loading time problems etc.). A website must be able to load in users' browser and must respond to user and in less than 120ms.

3.5. Logical database requirements

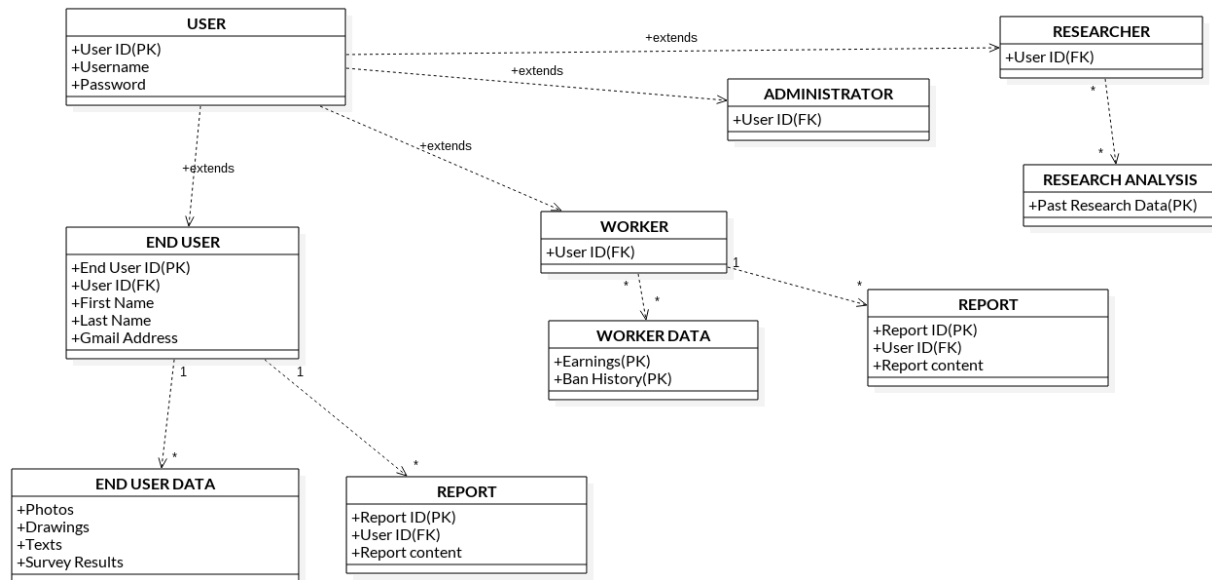


Figure 22: Logical Database Diagram

End User, Worker, Researcher and Admin tables are derived from User table. Admins can access all the tables. End users can not access any of them. All the information saved about End Users (relevant end user information, photos and drawings uploaded by end users) are stored in End User Data table. Both researchers, admins and workers can access this table but only the admins and workers edit the table. Also the reports composed by the End Users will be stored in Report table which is derived from End User table. This table is only accessible by administrators. All the information about with worker-related issues in the system is stored in Worker Data table and the reports composed by the Workers will be stored in Report table which is also derived from Worker table. Both table are only accessible by administrators. In the Research Analysis table all the information about past researches are stored. Researchers can access and edit this table.

3.6. Design constraints

The frameworks we have used to design the front-end and back-end parts of the system will strictly make programmers to obey the specific rules of layouts/templates etc. The system must comply with the all general and individual (of countries, regions etc.) specifications of conversational agent, application protocols and rules.

3.7. Software system attributes

Conversational reliability and availability are important concepts in Chorus system. System must be able to reliably address end users' questions. Since the it is active all the time servers will work 24/7, therefore some Failover mechanisms specific to the system is present. If a server goes down, then ongoing and new conversations will be processed by remaining active servers to ensure that the conversations remain uninterrupted. Furthermore while parties are in conversation with each other even if their computer crashes the system will save the chat history. To ensure availability, system servers may need to be geographically distributed and both conversational knowledge and transcript logs may need to be replicated. Also technical feedback system is available and the feedbacks will be checked and replied by administrators in very short time intervals.

Security is another important concept in the system. For account management and authorization processes Google will be responsible. For the internal mechanism of the conversation transmissions and storage, the system must provide at least the same level of security as Google. However the security levels that Chorus system must operate are greater than the Google since the conversations may contain highly private topics and may disclose critical personal information. Therefore the system conducts conversations over secure channels and it provides secure editing and promotion of content to numerous functionalities inside internal modules to the users. On the other hand, as a security precaution, server and database structure checks the integrity of every single file to be transmitted, which is hashed with SHA512, with checksum calculations of hashes and anything suspicious will be reported to the administrators by the system. For the user-related issues of the security precautions are taken by making users agree the agreement of security and privacy (End User Agreement etc.) in their sign up page, hence they are individually responsible for all their acts while interacting with the system. In addition, to ensure privacy purposes, any private end user information is kept in a secure area of the database and all conversations are confidential between end user and workers.

Portability is another crucial system attribute of Chorus. The system will operate on all the common operating systems (Windows, Linux, iOS Operating systems etc.) without requiring any specifications since the implementation of the main engine is done in Java and JVM is supported on these operating systems and the frameworks that are used in the back-end and front-end developments will not cause any particular environment problem.

Scalability is also an important concept in this project that requires a special attention. To offer a valid service to the end users, system must scale well. It must support tens of thousands of conversations and data transmissions each day. Capacity of the system servers and database is planned in accordance with the performance requirements determined earlier. Furthermore, the system is designed to address quick expansions of conversations on short time while crisis periods are ongoing. System is designed to dynamically increase all its resources needed to

handle unexpected additional conversation demand, since the expansion sizes are so difficult to anticipate.