

SOFTWARE REQUIREMENTS SPECIFICATION

Chorus Bot

Orhan Asan

2231264

Table of Contents

1. Introduction	4
1.1 Purpose of the Software	4
1.2 Scope	4
1.3 Product Overview	4
1.3.1 Product Perspective	4
1.3.1.1 System Interfaces	5
1.3.1.2 User Interfaces	6
1.3.1.3 Software Interfaces	10
1.3.1.4 Communication Interfaces	10
1.3.1.5 Memory Constraints	10
1.3.1.6 Operations	10
1.3.2 Product Functions	11
1.3.3 User Characteristics	12
1.3.4 Limitations	12
1.4 Definitions	13
2. References	14
3. Specific Requirements	15
3.1 External Interfaces	15
3.2 Functions	19
3.3 Usability Requirements	32
3.4 Performance Requirements	32
3.5 Logical Database Requirements	33
3.6 Design Constraints	34
3.7 Software System Attributes	34
3.8 Supporting Information	36
4. Verification	36
5. Appendices	36
5.1 Assumptions and dependencies	36
5.2 Acronyms and abbreviations	36

List of Tables

Table 1: Product functions	11
Table 2: Definitions	13
Table 3: Vote for final response.....	20
Table 4: Show response	21
Table 5: View system logs	22
Table 6: Post HIT.....	23
Table 7: Recruit workers	23
Table 8: Submit response candidate	24
Table 9: Vote for “working memory”	24
Table 10: Receive user message	25
Table 11: Show “working memory”	25
Table 12: Earn points.....	26
Table 13: Finish conversation	27
Table 14: Confirm Google user ID	27
Table 15: Log in.....	28
Table 16: Send user message	28
Table 17: Report inappropriate message	29
Table 18: Rate conversation	29
Table 19: Sign up	30
Table 20: View past message logs	30
Table 21: Authorize member	31
Table 22: Disallow member	31

List of Figures

Figure 1: Context Model.....	5
Figure 2: Desktop End-User Chat Interface	7
Figure 3: Mobile End-User Chat Interface	7
Figure 4: End-User Registration Interface	8
Figure 5: Worker Chatting Interface	9
Figure 6: External Interfaces Class Diagram	15
Figure 7: Use Case Model.....	19
Figure 8: Sequence diagram of “Vote for final response”	21
Figure 9: Sequence diagram of “Show response”	22
Figure 10: Logical Database Diagram.....	33

1. Introduction

1.1 Purpose

Introduced by a group of professors from Carnegie Mellon University, Chorus Bot will be a web-based and crowdsourced personal assistant whose mission is going to be achieving a rapid and ergonomic environment for solving problems that a person can come across in his/her daily life. A group of workers from all over the planet will try to solve problems in a collaborating manner (crowdsourcing), which will be the main source of rapid response environment of Chorus Bot.

1.2 Scope

- System will have a user chat interface where connection between user and worker(s) will be provided and user can ask his/her questions as text message, text message with speech recognition and photographs.
- System will contain a worker chat interface for each worker to interact with user and his/her current problems. Workers can submit response candidates and vote for them. They also can see important parts from user's previous conversations.
- System will contain interfaces for both system admins and IT staff. System admins will be able to manage members of system and also show message logs of a specified user. IT staff will be eligible to show latest errors in the system from their interfaces and also they can be notified by the system in case of an emergency.
- System will keep track of user conversations and members of system with a database interface. System admins can reach this database with the help of their interfaces in order to maintain system efficiency.
- System will have a recruitment interface for workers. The salaries of workers, instant available tasks and number of workers will be maintained through this interface. This interface will be autonomous, meaning that no external human interaction will be available unless exceptional situations.
- System will have a registration interface which user can specify his/her personal information and also give his/her consent to use Chorus Bot.

1.3 Product Overview

1.3.1 Product Perspective

Chorus Bot is not a part of a larger system but usually interacts with other well-known systems such as Google Hangouts and Amazon Mechanical Turk (MTurk). Google Hangouts is used for user interaction of the system whilst Amazon MTurk is used for worker interaction of the system. System also uses Google Forms for gathering registration forms from new coming users and authorize them to use Chorus Bot. In this context, Chorus Bot is not said to be interleaved with other systems but it uses other systems to supply communication through system.

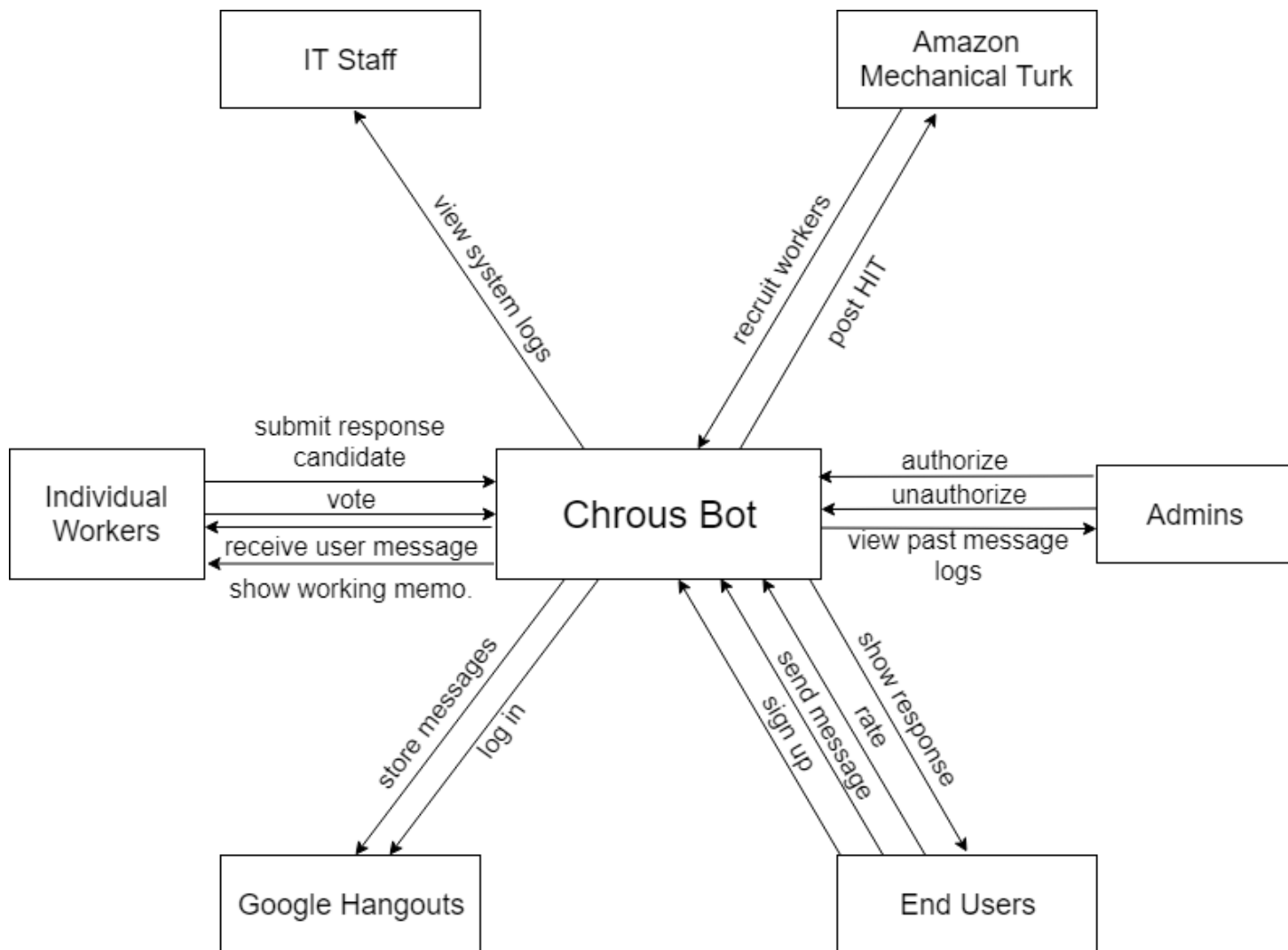


Figure 1: Context Model

In a more specific perspective, Chorus Bot cannot be divided into subsystems. However, system can be considered to consist of database that keeps track of members and chat logs, the user interfaces of users, workers, system admins and IT staff and also the system interfaces which provides interaction and communication of system components.

1.3.1.1 System Interfaces

Backend Voting Interface: This interface collects votes from workers throughout conversations for decisions such that voting for which responses to be sent and which user inputs should be remembered by Chorus Bot in order to be shown in later conversations. This interface keeps track of votes with workers' ID in case of any disputes occur against one of the workers.

Backend Error Management Interface: This interface is always active in case of any system failure can occur in whole system. It will be equipped with error logging functions and also error handling functions for small scale failures. There will be remote access option for IT staff to intervene to system rapidly. Basically, when this interface detects any failure or unexpected behavior in system, it writes down current situation in a detailed manner and also notifies IT staff (when needed).

Google Hangouts API: Chorus Bot uses Google Hangouts API to conduct user interaction side of the system. User can send text or photographs to workers to identify the question and/or problem, rate the conversation using a designated rubric or report inappropriate behavior of worker(s). Since there is a built-in voice recognition function in Google Hangouts, user can create inputs directly with his/her voice.

Amazon MTurk API: This API is used for creating Human Intelligence Task(s) (HITs), delivering assignments for workers, recruiting workers and managing the salaries of individual workers. Workers can also be blocked from system, when needed, or system admins can apply qualification restrictions for workers.

Google Forms API: User is required to sign up from Google Forms, where he/she must give his/her consent and personal information in order to start chatting with Chorus Bot. This data is then supplied to system admins through Google Forms API, whether they will be allowed to interact or not.

Database Management Interface: This interface is the main component of Chorus Bot since it keep tracks of every movement of worker(s) and user throughout every entire conversation. It has components to notify Amazon MTurk and Google Hangouts interfaces of workers and user, respectively. Every message of workers is kept here, no matter if they are shown to user or not. These messages are marked with their instant situation (for example, if it is in voting stage, this message is flagged as “inVoting”) and the additional information is supplied in other fields of database entry of the message. The error logs are also directed from Error Management Interface to here, in order to measure error percentage or up-time of the system.

1.3.1.2 User Interfaces

The user interfaces of Chorus Bot can be divided into 4 different type of users: main user, worker, system admin and IT staff. Main user interface will be easy to interact with system since our intention is keeping our system in a casual manner for end-users. As explained above, system has different interfaces for different purposes and/or users. More will be elaborated in External Interfaces section (3.1). This section will give an outermost view of these interfaces.

End-User Chat Interface: End-users, who are registered to the system, can interact with our system in a chatting manner through Google Hangouts. End-user can ask any type of questions or supply problems to the system to be solved and these inputs can be in the type of text message or photographs. The output, which will be given by our system, will be shown in this interface and also our system can provide evaluation functionality to end-user through Google Hangouts, such as rating conversation or reporting conversation.

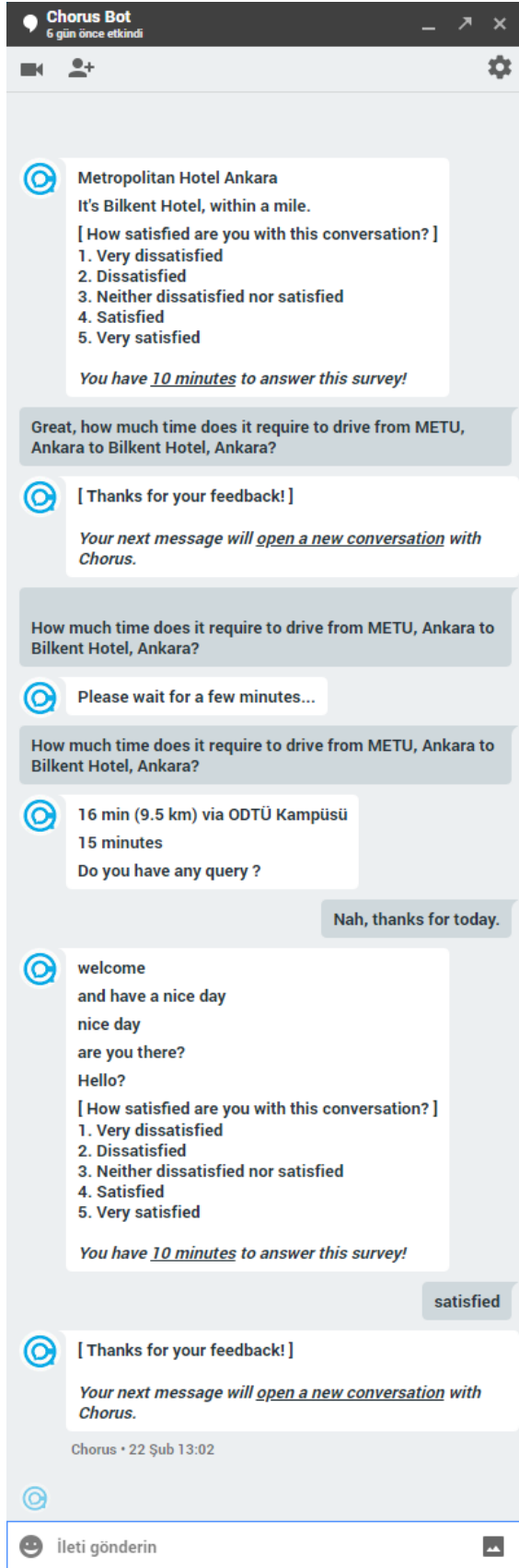


Figure 2: Desktop End-User Chat Interface

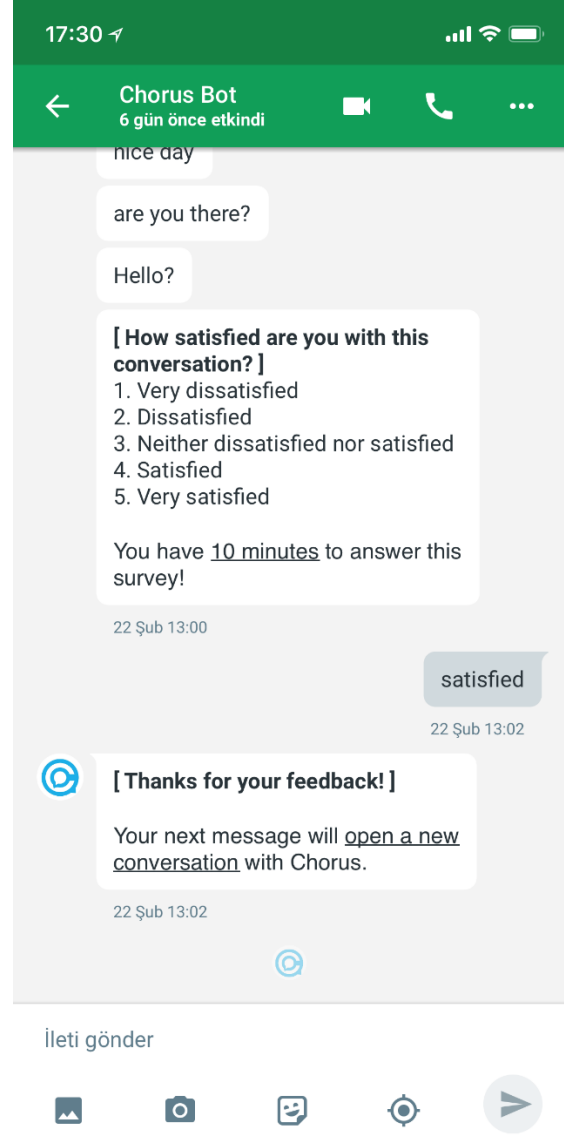
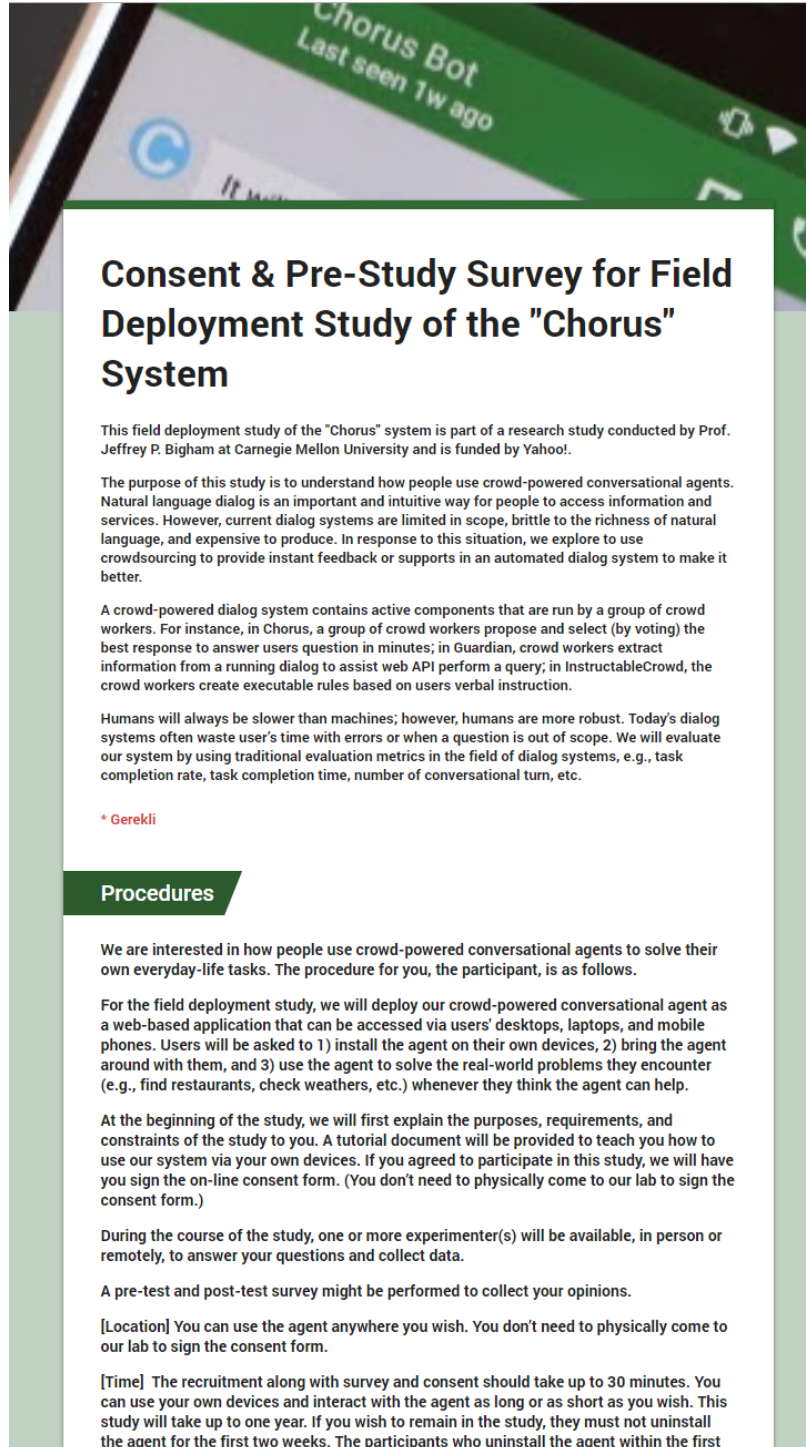


Figure 3: Mobile End-User Chat Interface

End-User Registration Interface: This interface is basically a Google Forms website which end-user is required to give his/her consent and personal information in order his registration to be taken into account by system admins. There is also a use policy and procedure expressed in this interface in order to keep end-user informed about potential consequences related to our system.



The image shows a mobile phone screen. At the top, a WhatsApp chat window is visible with a green header that says "Chorus Bot" and "Last seen 1w ago". Below the chat, a Google Form is displayed. The form has a white background with a green border. The title of the form is "Consent & Pre-Study Survey for Field Deployment Study of the 'Chorus' System". The form contains several paragraphs of text explaining the study, its purpose, and the procedures. At the bottom of the form, there is a green button labeled "Gerekli".

Consent & Pre-Study Survey for Field Deployment Study of the "Chorus" System

This field deployment study of the "Chorus" system is part of a research study conducted by Prof. Jeffrey P. Bigham at Carnegie Mellon University and is funded by Yahoo!.

The purpose of this study is to understand how people use crowd-powered conversational agents. Natural language dialog is an important and intuitive way for people to access information and services. However, current dialog systems are limited in scope, brittle to the richness of natural language, and expensive to produce. In response to this situation, we explore to use crowdsourcing to provide instant feedback or supports in an automated dialog system to make it better.

A crowd-powered dialog system contains active components that are run by a group of crowd workers. For instance, in Chorus, a group of crowd workers propose and select (by voting) the best response to answer users question in minutes; in Guardian, crowd workers extract information from a running dialog to assist web API perform a query; in InstructableCrowd, the crowd workers create executable rules based on users verbal instruction.

Humans will always be slower than machines; however, humans are more robust. Today's dialog systems often waste user's time with errors or when a question is out of scope. We will evaluate our system by using traditional evaluation metrics in the field of dialog systems, e.g., task completion rate, task completion time, number of conversational turn, etc.

* Gerekli

Procedures

We are interested in how people use crowd-powered conversational agents to solve their own everyday-life tasks. The procedure for you, the participant, is as follows.

For the field deployment study, we will deploy our crowd-powered conversational agent as a web-based application that can be accessed via users' desktops, laptops, and mobile phones. Users will be asked to 1) install the agent on their own devices, 2) bring the agent around with them, and 3) use the agent to solve the real-world problems they encounter (e.g., find restaurants, check weathers, etc.) whenever they think the agent can help.

At the beginning of the study, we will first explain the purposes, requirements, and constraints of the study to you. A tutorial document will be provided to teach you how to use our system via your own devices. If you agreed to participate in this study, we will have you sign the on-line consent form. (You don't need to physically come to our lab to sign the consent form.)

During the course of the study, one or more experimenter(s) will be available, in person or remotely, to answer your questions and collect data.

A pre-test and post-test survey might be performed to collect your opinions.

[Location] You can use the agent anywhere you wish. You don't need to physically come to our lab to sign the consent form.

[Time] The recruitment along with survey and consent should take up to 30 minutes. You can use your own devices and interact with the agent as long or as short as you wish. This study will take up to one year. If you wish to remain in the study, they must not uninstall the agent for the first two weeks. The participants who uninstall the agent within the first

Figure 4: End-User Registration Interface

Worker Chat Interface: Workers, who are recruited through Amazon MTurk, will interact with users in a collaborating manner with other workers from this interface. Workers can see user messages from this interface and can supply their answers to be voted by other workers and accepted through their interfaces. This interface also includes “Working memory” part which shows collected message logs from user’s past conversations with Chorus Bot and workers can also vote for important user messages to be marked as “important” and shown in “Working memory” section in conversations.

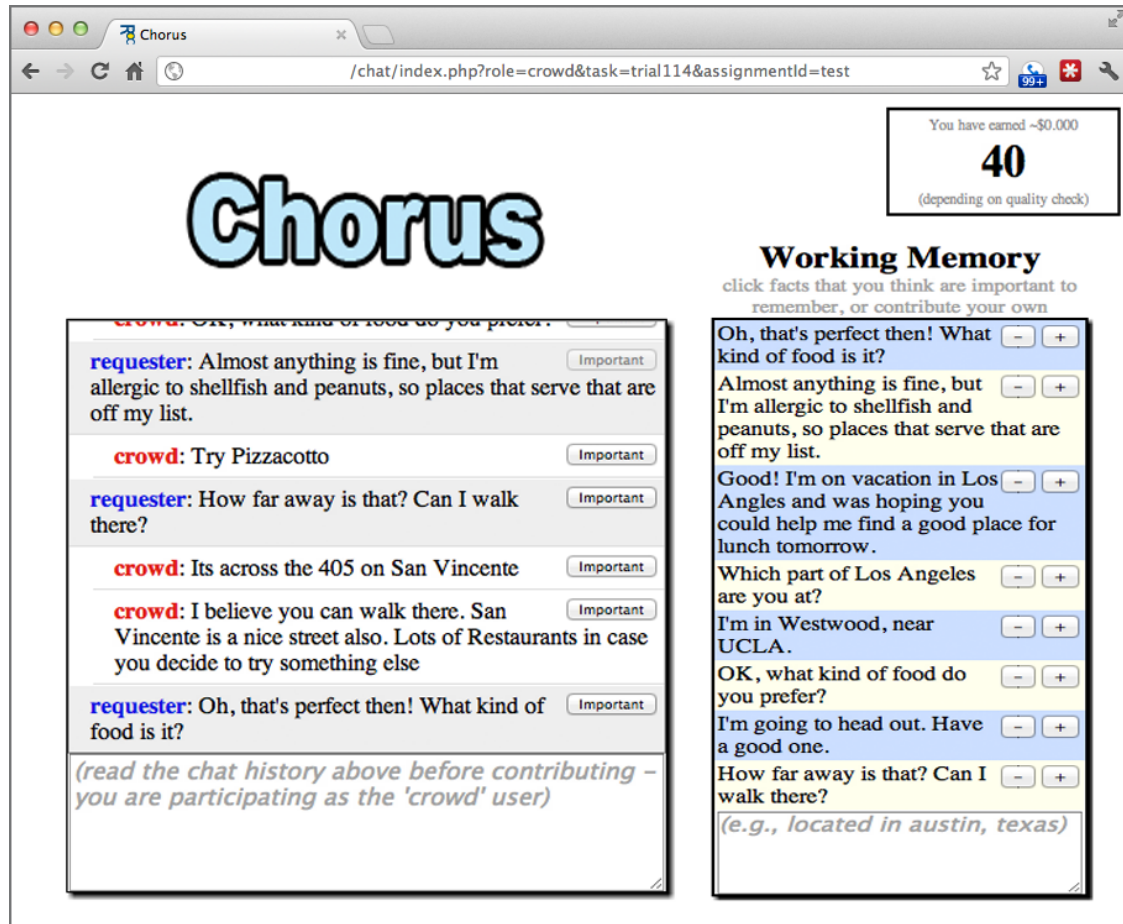


Figure 5: Worker Chatting Interface

System Admin Management Interface: In this interface, system admins can see the number of current conversations with their number of online workers. There are also notifications related to rating of conversations and reports filled by users for inappropriate manner of workers. System admins can see past message logs as an outer eye, where all kind of interactions are gathered from both user and workers side. System admins also have an account authorization interface implemented to their management interface, where they can see current applications to the system and authorize them to enter to the system. Also, from this interface they can disallow a current member of system (members can be end-users or IT staff) with their own initiatives.

IT Staff Management Interface: From this interface, IT staff will be available to see latest failures in the system and detailed logs about failures which will answer questions like when failures did happen, where system came across with them, which components

were affected, is failure still exist. This interface will have an always-on notification function that keeps staff updated about system in case of any critical failure.

1.3.1.3 Software Interfaces

Database: Chorus Bot uses database to store past message and error logs related to system as well as user IDs and personal information of registered users. Staff and system admins' authentication data is kept under different portion of Chorus Bot's database.

Operating system: Operating system would be server distributions of Linux and/or Windows.

1.3.1.4 Communications Interfaces

Chorus Bot uses HTTP/HTTPS for providing connection of workers between Amazon MTurk and its backend. HTTPS is a crucial need since authentication of workers are supplied from Amazon MTurk and security should be provided in Chorus Bot as HTTPS, so that system can avoid any over-payment issue or security breach.

1.3.1.5 Memory Constraints

Memory is not a huge issue for Chorus Bot since main interaction is provided by Amazon MTurk and Google Hangouts. However, system should have enough memory to sustain basic database operations and hosting web sites of workers.

1.3.1.6 Operations

The operations provided by Chorus can be divided into:

User operations:

- Sign up
- Log in
- Send user message
- Show response of workers
- Report inappropriate message
- Rate conversation

Worker operations:

- Receive user message
- Show "working memory"
- Submit response candidate to user
- Vote for final response
- Vote for "working memory"
- Earn points
- Finish conversation

Staff operations:

- View system logs (only for IT staff)
- Authorize member (only for system admins)
- Disallow member (only for system admins)
- View past message logs (only for system admins)

System operations:

- Post HIT
- Recruit workers
- Confirm Google user ID

All of these operations will be covered more deeply in Functions section (3.2).

1.3.2 Product Functions

Functionalities (use cases) of Chorus Bot is summarized as what they do in below. More advanced and detailed version can be found in Functions section (3.2) with their complete description tables.

Function	Summary
View system logs	Shows system related error logs to IT staff in IT Staff management interface.
Sign up	Lets user register to system through Google Forms.
Log in	Lets user log in if he/she sends his/her message after a long time.
Send user message	Lets user send message through Google Hangouts to be answered by workers.
Show response of workers	Shows message given by workers in user's chatting interface.
Report inappropriate message	Lets user report any toxic and abnormal behavior to the system admins.
Rate conversation	Lets user rate the previous conversation with a designated rubric.
Receive user message	Shows user message in workers' chatting interfaces.
Show "working memory"	Shows past user messages marked as "important" in user message database.
Submit response candidate	Lets worker submit his/her answer in order message to enter the process of becoming final response.
Vote for final response	Lets workers vote for an answer that they find suitable in order message to be final response

Vote for working memory	Lets workers vote for a user message to be marked as “important” in user message database in order to be shown in “working memory” section.
Earn points	Lets workers earn points based on their interactions with user.
Finish conversation	Finishes conversation if there is a consensus between workers or user is idle for an amount of time.
Authorize member	Lets system admin give permission to registered users or IT staff to use Chorus Bot.
Disallow member	Lets system admin forbid authorized users or IT staff to use Chorus Bot.
Post HIT	Creates HIT in order to recruit workers for Chorus Bot.
Recruit workers	Supplies a worker pool for continuity of Chorus Bot.
Confirm Google user ID	Lets system match user IDs from member database and Google Hangouts.

Table 1: Product functions

1.3.3 User Characteristics

The targets users of Chorus Bot system can be divided into three parts as non-staff members (end-users and workers), system admins and IT staff. System admins should have basic computer skills as they are required to interact with system admin UI. On the other hand, IT staff should expertise in their fields and they also know to handle IT staff UI. End-users should have basic knowledge about Google Hangouts, like how to log in or how to chat through Google Hangouts. Workers should be familiar with Amazon MTurk, like how to register, how to get payment or how to attend a HIT. Workers also know how to use their chatting interface, for this reason a tutorial will be supplied for a new coming worker.

1.3.4 Limitations

Regulatory policies: Personal information of users should not be released to public. Google user ID and e-mails of users will be kept encrypted outside the backend of the system.

Hardware limitations: Since processing time is not an important aspect in Chorus Bot, hardware should provide high bandwidth to conduct real time messaging between workers and users.

Interfaces to other applications: Chorus Bot should be compatible with Amazon MTurk, Google Hangouts and Google Forms in order to accomplish certain tasks provided in this document.

Parallel operation: Parallelization is recommended however it is not a must. System must be capable of serving to more people than average number of people that uses Chorus Bot daily.

Audit and Control functions: Account of workers and users are controlled by Amazon MTurk and Google Hangouts, therefore there is no need for any control or audit functions for them. On the other hand, system admins and IT staff interfaces should be equipped with such functions in order to minimize security breach and system failure probability.

Higher-order language requirements: System should be written in multi-platform object-oriented programming languages such as Java or Ruby to provide a compact design. In case of system failures and under certain constraints, operating system changes can be an obligation and these language requirements will come handy in these conditions.

Signal handshake protocols: HTTPS will be required interaction between Chorus Bot and Amazon MTurk, Google in account authorization processes. HTTPS will also be supplied in workers' interface to encrypt data transfer to avoid stolen personal information. Telnet or SSH connection will be provided to IT staff for situations where HTTP (Staff interface) is not possible. TCP will be used for database connection.

Quality requirements: Reliability and security is two most important priorities of Chorus Bot. System must be able to back up the databases once a day to separate HDDs in order to handle total system failures. Penetration tests must be conducted every week to minimize outcomes of security breaches.

Criticality of application: System failure is not life threatening for anybody, so Chorus Bot cannot be counted as a critical system. Furthermore, reliability of system is not needed to be high, excluding conservation of user and system data.

Safety and security considerations: Since worker and user security is controlled by Amazon MTurk and Google, there is no action needed to be taken in that side. However, system admin and IT staff account protection should be one of the most important priorities. In addition to that, database should be immune to any type of attacks or failures.

Physical/mental considerations: Anyone can use computer and/or mobile phones and has his/her consent to use system can use Chorus Bot, as long as he/she doesn't use system in a disruptive manner.

1.4 Definitions

Term	Definition
Database	A PostgreSQL database
Amazon MTurk	Amazon Mechanical Turk (https://www.mturk.com/)
API	Application programming interface
AUTHDATA	Authentication data
HIT	Human intelligence task
UI	User interface
COTS	Commercial off the shelf
ID	Unique number to identify a person

Table 2: Definitions

2. References

This document is written with respect to IEEE 29148-2011 standard:

IEEE. (2011, December 1). 29148-2011 - ISO/IEC/IEEE International Standard
- Systems and software engineering -- Life cycle processes --Requirements engineering.
Retrieved from <http://ieeexplore.ieee.org/document/6146379/> on March 12, 2018. doi:
10.1109/IEEESTD.2011.6146379

Other sources:

Bigham, J. P., & Huang, T. (2017). A 10-Month-Long Deployment Study of On-Demand Recruiting for Low-Latency Crowdsourcing. *In 5th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2017)*. Quebec City, Canada.

Lasecki, W. S., Wesley, R., Nichols, J., Kulkarni, A., Allen, J. F., & Bigham, J. P. (2013). Chorus: A Crowd-Powered Conversational Assistant. *In ACM Symposium on User Interface Software and Technology (UIST 2013)*. St. Andrews, UK.

3. Specific Requirements

3.1 External Interfaces

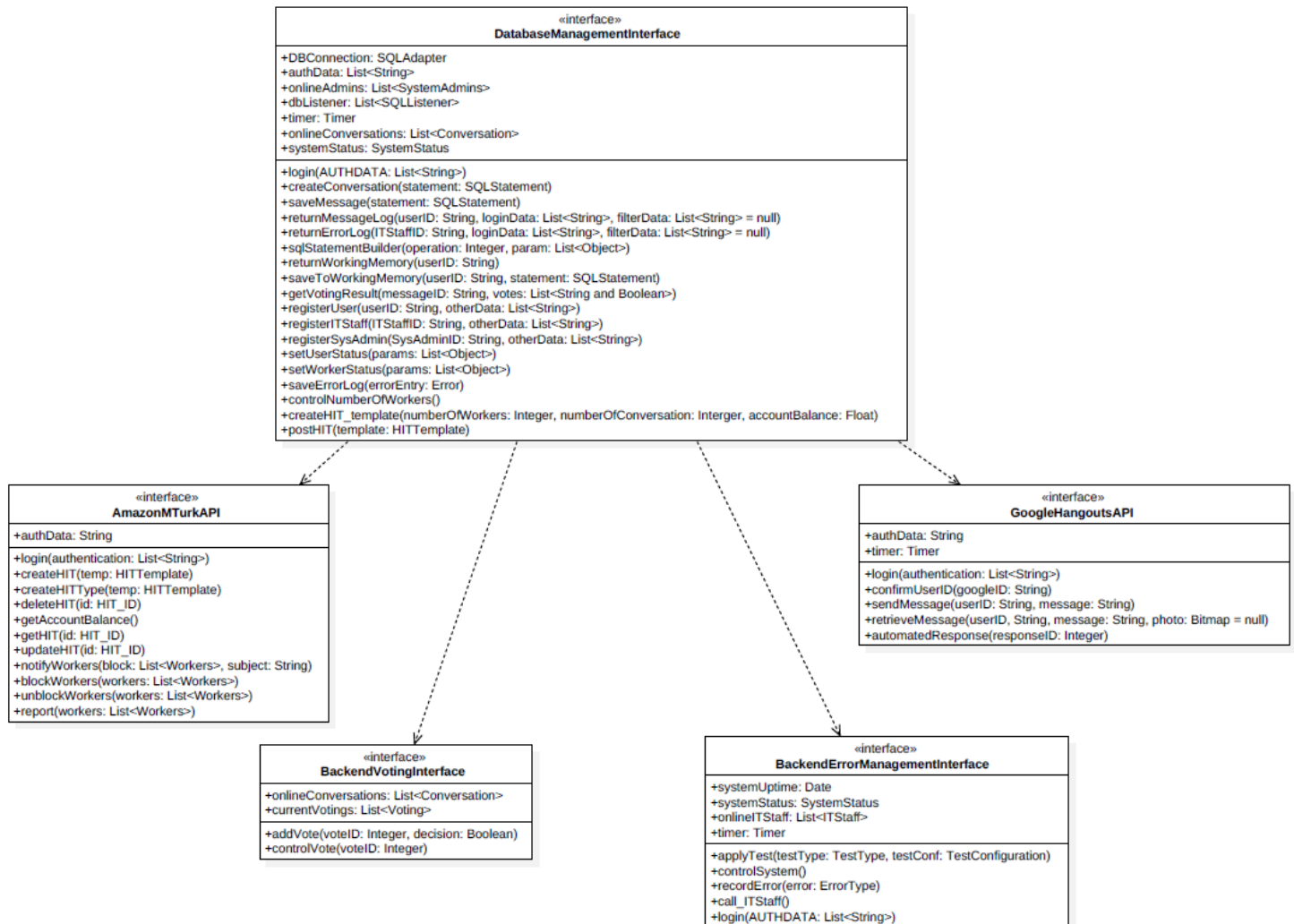


Figure 9: External Interfaces Class Diagram

3.1.1 End-User Chat Interface

req01: If connection is lost between user and system due to user-related problems, this interface shall warn user about this issue and remind them to check their internet connection.

req02: If user tries to reach unsupported features by system, this interface shall decline this request immediately and warn user about unavailability of this feature.

req03: This interface shall welcome first time user with an announcement that warns user to register to this service whether the user is authorized or not/registered or not.

req04: If user attempts to rate a conversation with an unsupported scheme, this interface shall inform user about how to rate a

conversation with a descriptive manner and wait user to input his/her rating one more time.

req05: If user request is in correct manner, this interface shall route this request to Database Management Interface with user message flag.

req06: This interface shall measure response time of user during conversation and compare these measurements in case of any bot uses. If so, user shall be blocked from the system and informed about this issue and how to object this decision and also System Admin Management Interface shall be informed.

3.1.2 End-User Registration Interface

req07: If user does not enter his/her demanded personal information or accept to give his/her concept, this interface shall not continue to registration process and warn user in order to enter and/or accept these issues.

req08: This interface shall conduct survey about daily messaging habits during registration.

req09: This interface shall warn users about risks, procedures, requirements and other type of legal issues of Chorus Bot before proceeding to registration.

req10: This interface shall direct registration data to Database Management Interface with member registration flag if all of the inputs are correct and complete.

3.1.3 Worker Chat Interface

req11: This interface shall warn worker if he/she has not conducted any operation for 10 minutes (operation definition includes all functions that worker can use in this interface). If worker does not respond to the warning, worker should be dismissed from the task and this interface shall return worker to the waiting page.

req12: This interface shall inform worker when he/she reaches a designated amount of points and congratulate them.

req13: This interface shall not let worker to enter his/her personal information and detect these type of information. If this sort of messages is detected, this interface shall report this issue to System Admin Management Interface, deduct points because of this incident and inform worker about that.

req14: This interface shall be responsible for detecting bot activities. If any botting incident occur, this interface shall immediately block this worker from system, inform System Admin Management Interface and inform worker about how to object to this decision.

req15: This interface shall notify worker when a user message arrives to this interface, anyone from other workers opens a

message for vote and a user message is added to “working memory”.

req16: Using sentences that impose verbal abuse and toxicity shall be strictly punished. Therefore, this interface shall be capable of tracking conversation with respect to restricted words dictionary. If any of these words is found when a worker tries to send it, he/she shall be blocked from system after first warning.

3.1.4 System Admin Management Interface

req17: This interface shall notify system admin when reports of incidents (such as verbal abuse or detection of bots) come from other interfaces. The notification shall be done with both sending e-mail and text messages from mobile phone as well as updating UI that system admin uses.

req18: System admin shall see users’ last message logs and personal information when he/she is in “disallow member” component of this interface.

req19: This interface shall send e-mails to both authorized and disallowed users at the time of system admin finalizes the procedure. This e-mail shall notify user/IT staff about the process and contain cause of dismissal for disallow member procedure.

req20: This interface shall show real-time performance and user statistics to system admin.

req21: This interface shall enable system admin to manage Amazon MTurk account of the system. System admin shall be able to do most of the operations related to Amazon MTurk through this interface.

3.1.5 IT Staff Management Interface

req22: This interface shall notify IT staff when a failure and/or error occurs in the system through sending e-mail or mobile text messages. IT staff shall be notified by sending voice mails in addition to above precautions in case any critical issue. In addition, this interface shall notify System Admin Management Interface in case of any critical failure.

req23: IT staff shall be eligible to see performance and usage statistics of each component of the system.

req24: This interface shall have an internal communication component to establish connection between all IT staff.

3.1.6 Backend Voting Interface

req25: This interface shall store all of the voted messages in its internal database. These messages shall be found with its initiator’s worker ID, the worker IDs of workers that voted for and against it.

req26: This interface shall return voting details when it is requested

by System Admin Management Interface.

req27: This interface shall allow voting approval equation to be changed by System Admin Management Interface when it is requested and write this change to system logs.

req28: This interface shall keep track of amount of votes given by certain worker and shall report it to Database Management Interface to update worker's balance.

3.1.7 Backend Error Management Interface

req29: This interface shall categorize each error and/or failure depending on where it is occurred and what might be the cause of it. Then, this interface shall pass this entry to Database Management Interface in order to save it.

req30: This interface shall run predesigned stress and penetration tests on the system in maintenance hours and report their results to IT staff.

req31: This interface shall freeze all user and worker chat operations during maintenance hours.

req32: This interface shall provide remote access to IT staff. Authorization shall be delivered by Database Management Interface and it shall use SSL for authentication and Telnet for remote access.

3.1.8 Google Hangouts API

req33: This API shall provide Google user ID to system when required.

req34: This API shall be able to provide user messages of past 30 days in case of any corruption of databases.

req35: This API shall report user marked as bot by system to Google Hangouts. The detailed message logs shall be provided while reporting occurs and it shall return case ID to Database Management Interface to keep record of it.

3.1.9 Amazon MTurk API

req36: This API shall provide Amazon MTurk worker ID when it is required. It shall create worker records which consist of worker ID and his/her total earned points and pass these records to Database Management Interface.

req37: This API shall inform Amazon MTurk if worker is flagged as bot and shall provide detailed interaction history of worker as evidence.

req38: This API shall allow system to post HITs which consist of HTML pages independent from Amazon MTurk.

3.1.10 Google Forms API

req39: This API shall be eligible to serve statistical data of surveys conducted while user is signing up.

3.1.11 Database Management Interface

req40: This interface shall categorize each entry from which component it comes. Then, it shall direct them to appropriate databases and tables.

req41: This interface shall be responsible of account management and login of system admins and IT staff.

3.2 Functions

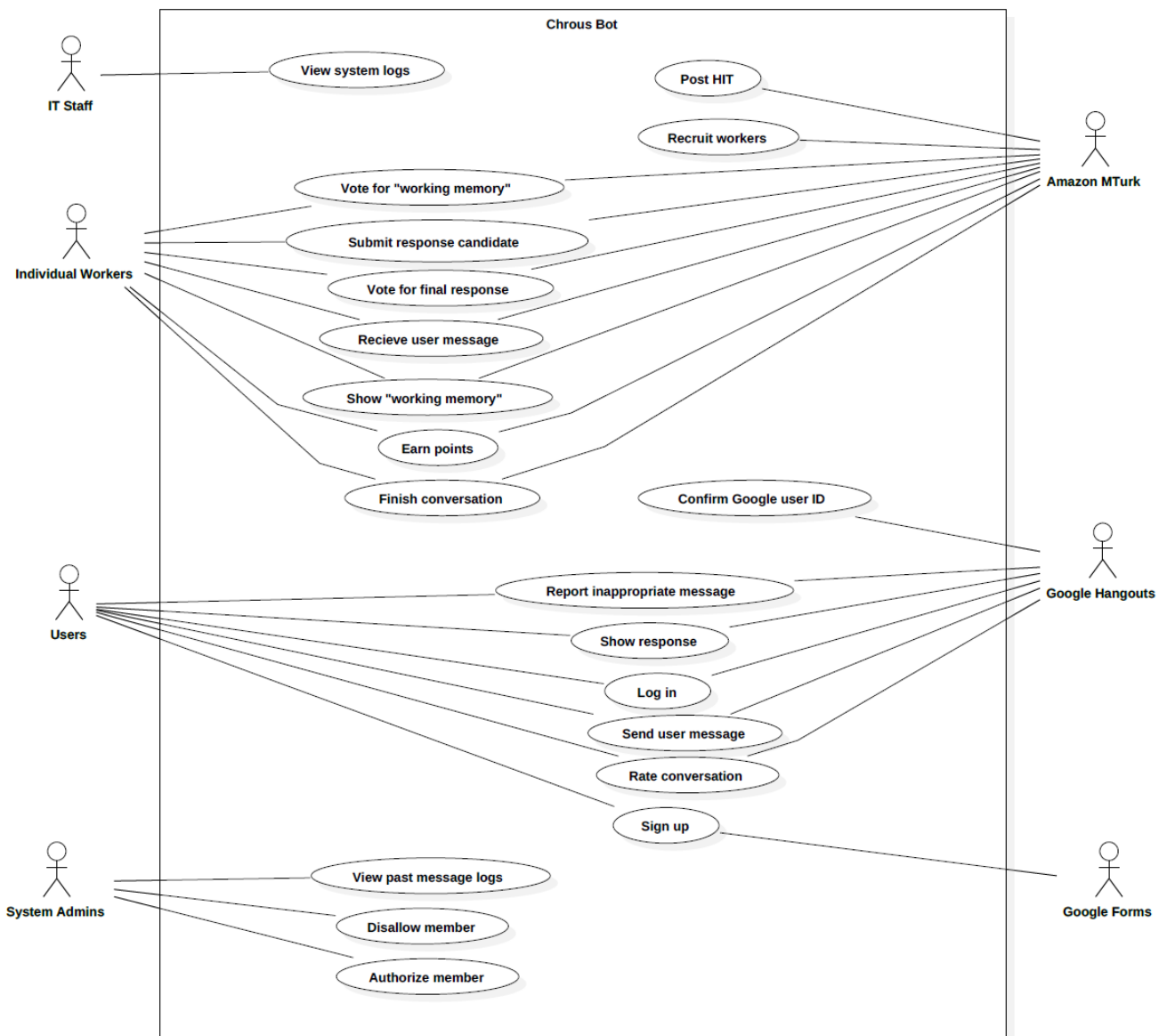


Figure 7: Use Case Model

Use case name	Vote for final response
Actors	Individual workers, Amazon MTurk
Description	If a worker submits a new response candidate, the Chorus system opens this answer for voting. If enough positive votes (compared to negative votes) is given by workers, system marks this response as final response.
Data	Number of positive and negative votes given, number of active workers, submitted text message by worker, worker ID
Preconditions	Worker number must be at least one to proceed.
Stimulus	Chorus Bot is notified when user message database has a new worker entry.
Basic Flow	<p>Step 1 – Chorus Bot is notified by user message database.</p> <p>Step 2 – The response candidate of worker is obtained from message database.</p> <p>Step 3 – Chorus Bot passes the submitted message to other workers' Amazon MTurk interfaces.</p> <p>Step 4 – Workers use their votes positively or negatively, and these votes are transmitted into backend voting interface.</p> <p>Step 5 – An agreement has been reached amongst workers i.e. the threshold is reached in voting.</p> <p>Step 6 – Message is marked as final amongst all workers' interfaces and saved to user message database.</p>
Alternative Flow #1	<p>Step 5 – If there is no positive consensus on this candidate, Chorus Bot waits for another candidate or consensus to be reached.</p> <p>Step 6 – If there is no consensus achieved for a designated amount of time, Chorus Bot informs user about this issue.</p>
Alternative Flow #2	Step 5 - If the worker who submitted the message is disconnected from conversation, the message will be removed from all workers' interfaces.
Exception Flow	If Amazon MTurk interface cannot provide an output to Chorus Bot due to connectivity problems, this is saved to Error log file.
Postconditions	Chorus Bot marks candidate as final response.

Table 3: Vote for final response

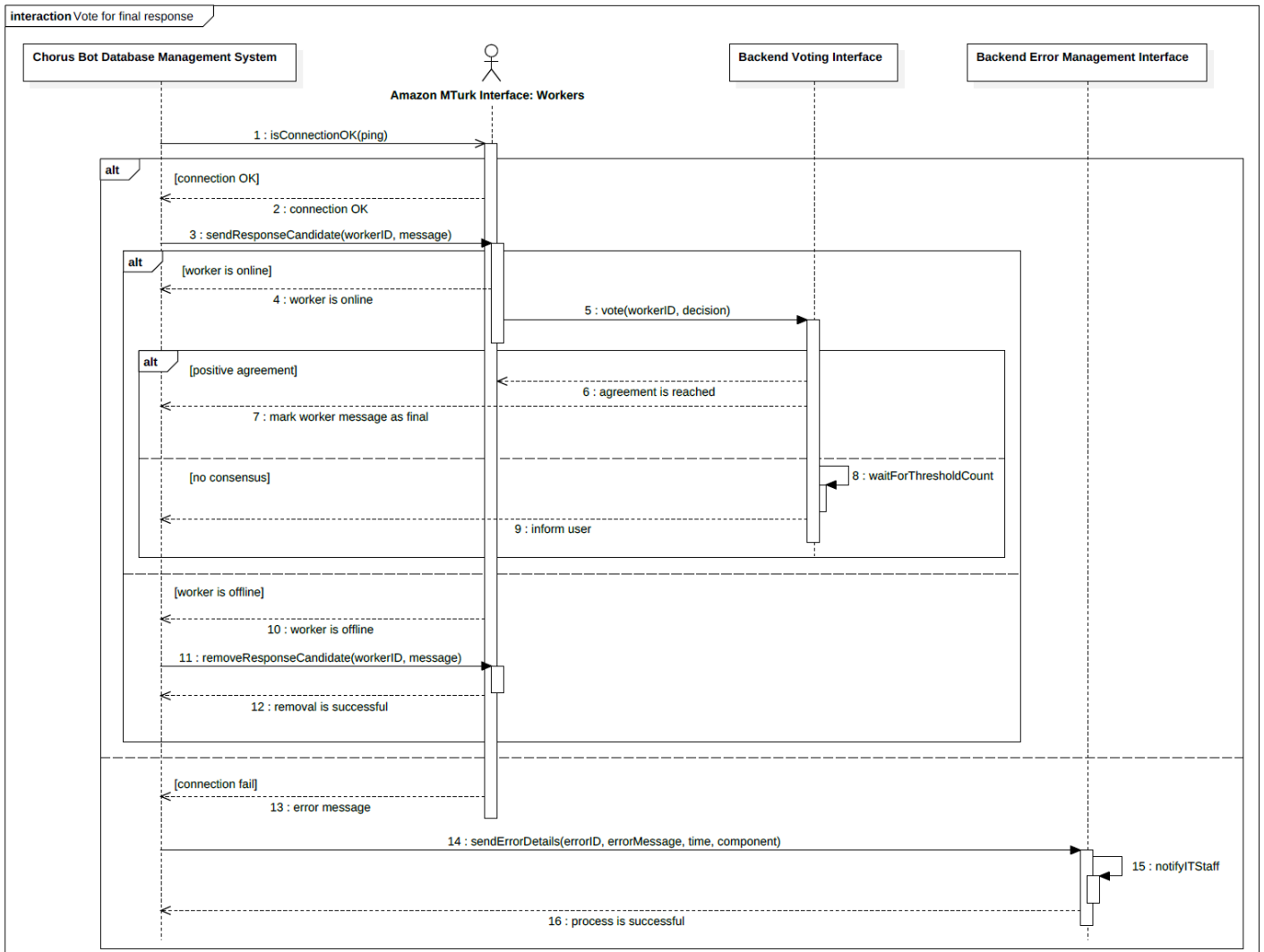


Figure 8: Sequence diagram of “Vote for final response”

Use case name	Show response
Actors	User, Google Hangouts
Description	If a new marked response which is sent by workers is found in user message database of Chorus Bot, system transfers this response to Google Hangouts in order to show message to user by Google Hangouts API.
Data	Text message voted as final by workers and marked by system.
Preconditions	-
Stimulus	A database listener notifies system when a new response entry is found in user message database.
Basic Flow	<p>Step 1 – Database listener notifies Chorus Bot.</p> <p>Step 2 – Chorus Bot uses Google Hangouts API in order to login to Google Hangouts.</p> <p>Step 3 – Chorus Bot passes message to Google Hangouts API.</p>

	Step 4 – Google Hangouts API communicates with Google Hangouts and passes the message.
Alternative Flow	Step 3 – If login is failed, system tries to log in for a designated amount of time. Step 4 – If there is no login after Step 3, system notifies IT staff.
Exception Flow	If an error thrown out by Google Hangouts API, it is written to System error log file and IT Staff is notified.
Postconditions	Message is shown in user’s Google Hangouts interface

Table 4: Show response

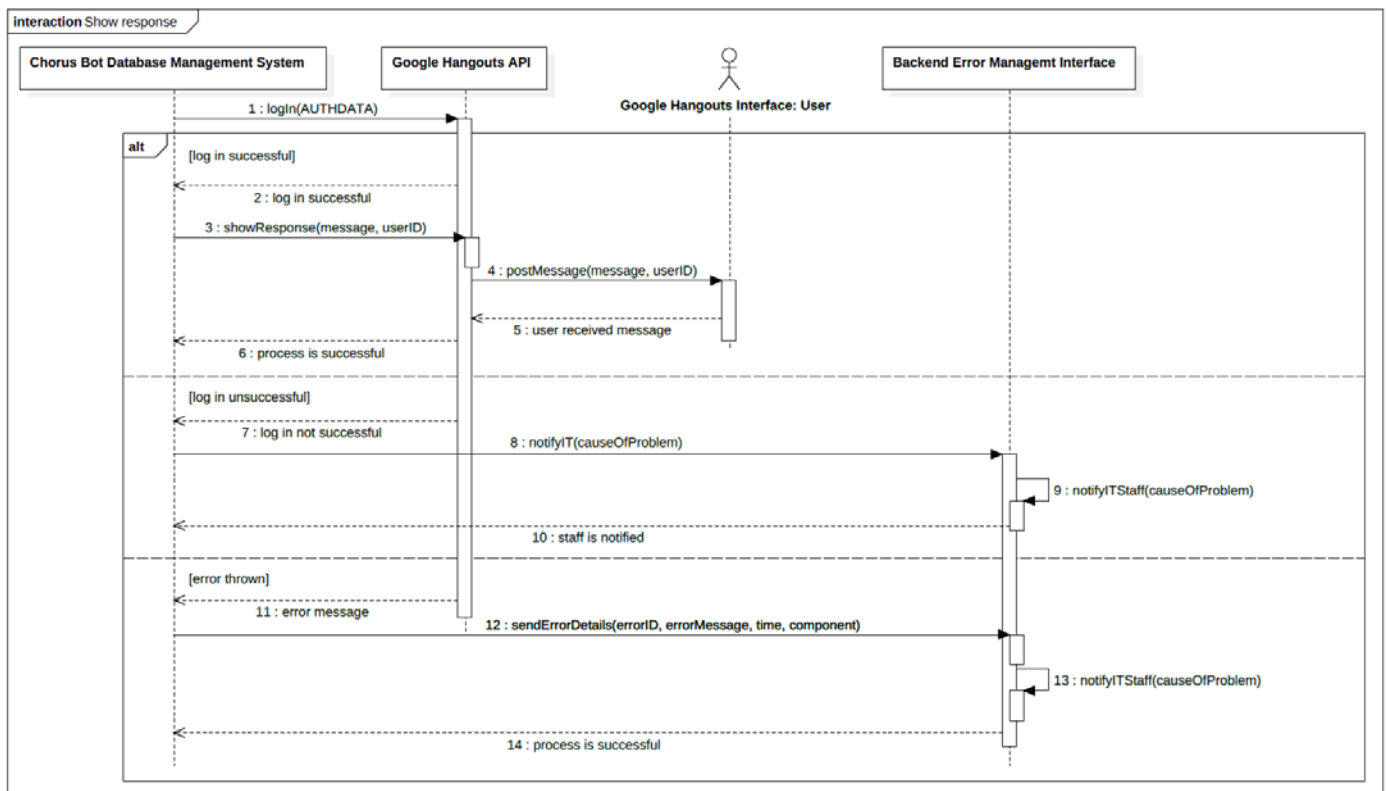


Figure 9: Sequence diagram for “Show response”

Use case name	View system logs
Actors	IT staff
Description	IT staff can list the system logs of Chorus Bot and logs can be sorted by their date of formation and the components of the system where the events take place. These logs include technical details of Chorus Bot system.
Data	System logs
Preconditions	IT staff must be authorized to read system logs.
Stimulus	One of IT staff tries to obtain system logs.
Basic Flow	Step 1 – System logs retrieval request is reached to Chorus Bot.

	Step 2 – Chorus Bot reaches database of the system and gathers logs. Step 3 – Chorus Bot passes the logs to IT staff's interface.
Alternative Flow	-
Exception Flow	If database connection is lost, system notifies IT staff's interface.
Postconditions	All of the system logs are shown in IT staff's interface.

Table 5: View system logs

Use case name	Post HIT
Actors	Amazon MTurk
Description	Chorus Bot constantly checks number of workers and number of conversations in order to properly distribute workload. When the amount of workers drops below a designated number depending on number of online conversations, system begins to introduce new works to HIT market.
Data	Number of workers, number of online conversations and average response time
Preconditions	-
Stimulus	Number of workers is below a certain level.
Basic Flow	Step 1 – A listener reminds system about insufficient number of workers. Step 2 – A HIT template is created depending on number of workers needed and how many conversations is ongoing. Step 3 – This template is sent to Amazon MTurk by its API. Step 4 – Amazon MTurk API returns task placed as successful.
Alternative Flow	-
Exception Flow	If connection is lost, it is written to system logs. Depending on balance of work load, IT staff can be notified.
Postconditions	A HIT is eligible on Amazon MTurk.

Table 6: Post HIT

Use case name	Recruit workers
Actors	Amazon MTurk
Description	Chorus Bot is notified by Amazon MTurk when a worker is eligible to do tasks. Chorus Bot starts hiring procedure which consists of showing a tutorial based on use of worker interface and evaluation of tutorial to complete recruitment process.
Data	ID of the worker, qualifications of the worker
Preconditions	Worker must have completed a required amount of tasks on Amazon MTurk in advance to participate.

Stimulus	A worker accepts the task.
Basic Flow	Step 1 – Worker reaches to landing page. Step 2 – Worker asked to do a tutorial in order to understand how system works. Step 3 – After tutorial is done, worker is put in a waiting list. Step 4 – After a conversation is found for worker to participate, worker is taken to conversation page.
Alternative Flow	Step 2 – If worker did tutorial, he/she automatically lands to waiting page. Step 3 - After a conversation is found for worker to participate, worker is taken to conversation page.
Exception Flow	-
Postconditions	Worker starts to participate in the conversation.

Table 7: Recruit workers

Use case name	Submit response candidate
Actors	Individual workers, Amazon MTurk
Description	If a worker decides to answer a message, an agreement should be reached amongst all workers in order message to be shown to user. Therefore, this message should be opened to voting.
Data	Submitted text message by worker, worker ID
Preconditions	-
Stimulus	Amazon MTurk interface notifies Chorus Bot when someone in workers on Amazon MTurk submit a response candidate for last user message.
Basic Flow	Step 1 – Amazon MTurk notifies Chorus Bot. Step 2 – The response candidate of worker is obtained from his/her Amazon MTurk interface. Step 3 – This message is written to message database with his/her worker ID.
Alternative Flow	Step 4 – If the worker is disconnected from conversation, the message will be removed from message database.
Exception Flow	If Amazon MTurk interface cannot provide an output to Chorus Bot due to connectivity problems, this is saved to Error log file.
Postconditions	Chorus Bot notifies message listener.

Table 8: Submit response candidate

Use case name	Vote for “working memory”
Actors	Individual workers, Amazon MTurk

Description	If a worker finds a response given by user important, he/she can vote for this message to be saved into a database in order to be shown in future conversations of this user.
Data	Number of positive and negative votes given, number of active workers, submitted text message by user, worker ID, user ID
Preconditions	-
Stimulus	Chorus Bot is notified when a worker clicks “important” button embedded to a user message.
Basic Flow	Step 1 – Chorus Bot is notified by button click. Step 2 – System waits for other workers to vote for this message as important until end of conversation. Step 3 – If all of the votes passes a threshold level, this user message is saved into a working memory database with user ID.
Alternative Flow	-
Exception Flow	-
Postconditions	“Working memory” listener is notified.

Table 9: Vote for “working memory”

Use case name	Receive user message
Actors	Individual workers, Amazon MTurk
Description	If the user sends a message, this message is passed to all workers by an individual user message database.
Data	Submitted text message by user, user ID
Preconditions	At least one worker is active.
Stimulus	A message listener notifies Chorus Bot.
Basic Flow	Step 1 – Chorus Bot is notified about new user message. Step 2 – Submitted message is taken from user message database. Step 3 – Message is passed to all workers’ Amazon MTurk interfaces.
Alternative Flow	-
Exception Flow	-
Postconditions	All workers can read user message.

Table 10: Receive user message

Use case name	Show “working memory”
Actors	Individual workers, Amazon MTurk
Description	When a worker connects the conversation, “working memory” listener is stimulated to show previous user messages marked as

	important. This listener notifies Chorus Bot constantly which means “working memory” can change constantly depending on current conversation.
Data	Marked important responses, user ID
Preconditions	At least one worker is active.
Stimulus	Chorus Bot notifies “working memory” listener.
Basic Flow	Step 1 – “Working memory” listener starts to collect previous user messages marked as important. Step 2 – These messages are passed to all workers’ Amazon MTurk interfaces.
Alternative Flow	Step 2 – If this conversation is user’s first conversation, all workers are notified about this issue in their Amazon MTurk interface.
Exception Flow	-
Postconditions	All workers see past important user messages.

Table 11: Show “working memory”

Use case name	Earn points
Actors	Individual workers, Amazon MTurk
Description	When a worker contributes to conversation in designated ways (for example voting for final responses, marking messages as important and submitting responses), an amount of money will be given to worker in order to increase contribution to the system.
Data	Type of worker action, worker ID
Preconditions	-
Stimulus	Amazon MTurk interface notifies Chorus Bot when a worker does a certain action.
Basic Flow	Step 1 – Chorus Bot decides which type of action worker committed to. Step 2 – System adds balance to worker ID parallel to designated rewarding scheme. Step 3 – Worker balance is recorded to worker database Step 4 – Amazon MTurk is notified about this issue, so that worker can be paid with updated salary.
Alternative Flow	-
Exception Flow	If new balance cannot be passed to Amazon MTurk, system admins will be notified.
Postconditions	Balance of worker is updated.

Table 12: Earn points

Use case name	Finish conversation
Actors	Individual workers, Amazon MTurk
Description	When a worker clicks for “Initiate voting for finishing conversation” button in his/her interface, a pop-up will appear on other workers’ interface to vote for. If a threshold is reached, procedure of finishing conversation will be started. Otherwise, this failure will be written in user message logs with a different indicator and also worker ID will be written for further inspections.
Data	Worker ID
Preconditions	At least one worker is active.
Stimulus	Amazon MTurk interface notifies Chorus Bot.
Basic Flow	Step 1 – Chorus Bot notifies other workers’ interfaces by a pop-un and begins for voting procedure. Step 2 – When a certain threshold is reached for voting, rating component will be called from Chorus Bot in order to finalize conversation
Alternative Flow	Step 2 – If a certain threshold is not reached for certain amount of time, this attempt will be written to user message logs with a different indicator. Step 3 – Worker ID will be reported to system admins.
Exception Flow	-
Postconditions	Amazon MTurk interfaces of all workers’ are returned to waiting page.

Table 13: Finish conversation

Use case name	Confirm Google user ID
Actors	Google Hangouts
Description	When a user log in into the system by sending first message to Chorus Bot through Google Hangouts, Chorus Bot confirms user ID whether user is authorized to interact with Chorus Bot system.
Data	User ID
Preconditions	User is started log in procedure in Google Hangouts.
Stimulus	“Log in” listener notifies system.
Basic Flow	Step 1 – Chorus Bot is notified by listener Step 2 – System checks if user ID is found in database and returns a welcome message.
Alternative Flow	Step 2 – If user is not found in the system, an information about how to sign up to the system will be returned.
Exception Flow	-
Postconditions	Google user ID is confirmed.

Table 14: Confirm Google user ID

Use case name	Log in
Actors	User, Google Hangouts
Description	When a user starts a chat with Chorus after a large amount of time, system firstly try to confirm credentials of user. If credentials are confirmed by “Confirm Google user ID” part, user is directed into conversation.
Data	User ID, message of user
Preconditions	-
Stimulus	Google Hangouts notifies Chorus Bot when a message reaches from user.
Basic Flow	Step 1 – Chorus Bot is notified by Google Hangouts. Step 2 – User ID is obtained by Google Hangouts API. Step 3 – Google ID of user is confirmed by relevant component. Step 4 – System passes message returned by relevant component.
Alternative Flow	Step 5 – If credentials are not confirmed by user database, inputs of user are ignored until user signs up and is authorized to interact.
Exception Flow	-
Postconditions	System is ready to chat with user.

Table 15: Log in

Use case name	Send user message
Actors	User, Google Hangouts
Description	If a user sends a message through Google Hangouts, firstly this message will be identified whether it is a request, reporting a concern or rating the conversation. Depending on type (reporting or rating), relevant components will be initiated. If message is a request, message will be saved to user message database and directed to workers.
Data	Text message of user, user ID
Preconditions	-
Stimulus	Google Hangouts notifies system when a new message is submitted by user.
Basic Flow	Step 1 – Google Hangouts notifies Chorus Bot. Step 2 – Chorus Bot uses Google Hangouts API in order to obtain user message. Step 3 – Chorus Bot handles message. Step 4 – If it is a request, it is saved into message logs with user ID.
Alternative Flow #1	Step 4 – If it is a report, report component is initiated.

Alternative Flow #2	Step 4 – If it is about rating conversation, rating component is initiated.
Exception Flow	If an error thrown out by Google Hangouts API, it is written to System error log file.
Postconditions	Chorus Bot is notified about the new entry in user message database.

Table 16: Send user message

Use case name	Report inappropriate message
Actors	User, Google Hangouts
Description	When a user finds a message is irrelevant, toxic and/or humiliating, he/she can report it to admins, so that the worker can be blocked from Chorus Bot after a proper investigation.
Data	User ID, worker ID, message(s) of worker(s)
Preconditions	A worker message is found in the conversation
Stimulus	Google Hangouts notifies Chorus Bot when a report request reaches from user.
Basic Flow	<p>Step 1 – Chorus Bot is notified by Google Hangouts.</p> <p>Step 2 – A message will be directed to user about what is the cause of concern.</p> <p>Step 3 – User input for cause of concerned are saved to report log along with whole conversation and worker ID(s).</p> <p>Step 4 – System admins are notified.</p> <p>Step 5 – Workers are also notified about the concern in order to supply awareness.</p>
Alternative Flow	-
Exception Flow	-
Postconditions	A success message about report procedure is shown to user.

Table 17: Report inappropriate message

Use case name	Rate conversation
Actors	User, Google Hangouts
Description	When the conversation seems to finished i.e. certain answers like “Have a nice day.” is received from workers or finishing conversation procedure is finalized by workers, system automatically send a message about rating conversation with predefined wordings. User can choose one of the words in order to rate conversation and send it through Google Hangouts.
Data	User ID
Preconditions	Conversation is marked as “finished”

Stimulus	Chorus Bot is notified when a conversation is marked as “finished”
Basic Flow	Step 1 – Chorus Bot is notified. Step 2 – A message will be directed to user about rating procedure. Step 3 – User input for satisfaction of user about conversation is saved to user message logs with satisfaction rating and also “end of conversation” indicator is saved to user message logs.
Alternative Flow	Step 4 – If satisfaction rating is low, conversation will be flagged for further inspection by system admins.
Exception Flow	-
Postconditions	A message will be sent to user to inform about conversation is closed.

Table 18: Rate conversation

Use case name	Sign up
Actors	User, Google Forms
Description	When a user decides to sign up for Chorus, he/she will supply his personal information through Google Forms and also his/her consent to attend Chorus Bot study. This information will be then passed to system admin interface of Chorus Bot.
Data	Google user ID, personal information and consent of user
Preconditions	-
Stimulus	User opens sign up page of Chorus.
Basic Flow	Step 1 – User fills the sign up form. Step 2 – After user clicks to “Submit” button, data will be sent to Google Drive account of Chorus Bot. Step 3 – Chorus Bot system checks regularly for new inputs in Google Forms through Google Forms API and notifies system admins interface.
Alternative Flow	-
Exception Flow	If an error thrown out by Google Forms API, it is written to System error log file.
Postconditions	User is ready to be authorized for conversation.

Table 19: Sign up

Use case name	View past message logs
Actors	System admins
Description	System admins can list the past message logs of a user whose ID is known for certain purposes and they can see all of the interactions of both user and workers throughout conversations. These logs are ordered by time.

Data	Past message logs, user ID
Preconditions	System admins must be logged in to their interfaces.
Stimulus	One of system admins tries to gather past message logs.
Basic Flow	Step 1 – Past message logs retrieval request is reached to Chorus Bot. Step 2 – Chorus Bot reaches database of the user message and gathers logs belonging to that user. Step 3 – Chorus Bot passes the logs to system admin’s interface.
Alternative Flow	-
Exception Flow	If database connection is lost, system notifies IT staff and system admin’s interface.
Postconditions	All of the past message logs of user are shown in system admin’s interface.

Table 20: View past message logs

Use case name	Authorize member
Actors	System admins
Description	System admins can authorize both IT staff and user in order them to interact with Chorus Bot. The authorization is processed through databases that belongs either IT staff of users.
Data	User ID or IT staff ID
Preconditions	System admins must be logged in to their interfaces.
Stimulus	One of system admins tries to authorize one person
Basic Flow	Step 1 – Authorize request is reached to Chorus Bot. Step 2 – Chorus Bot identifies whether this process involves IT staff or a user. Step 3 – If it involves user, given user ID will be marked as “authorized” in users database.
Alternative Flow	Step 3 - If it involves IT staff, given IT staff ID will be marked as “authorized” in staff database.
Exception Flow	If database connection is lost, system notifies IT staff and system admin’s interface.
Postconditions	Message will be shown to system admin which says “Authorization process is successful.”

Table 21: Authorize member

Use case name	Disallow member
Actors	System admins

Description	System admins can also disallow both IT staff and user in order to dismiss them from Chorus Bot. The process is similar to authorization, however this time members are flagged as “disallowed”.
Data	User ID or IT staff ID
Preconditions	System admins must be logged in to their interfaces.
Stimulus	One of system admins tries to disallow one person
Basic Flow	Step 1 – Disallow request is reached to Chorus Bot. Step 2 – Chorus Bot identifies whether this process involves IT staff or a user. Step 3 – If it involves user, given user ID will be marked as “disallowed” in users database.
Alternative Flow	Step 3 - If it involves IT staff, given IT staff ID will be marked as “disallowed” in staff database.
Exception Flow	If database connection is lost, system notifies IT staff and system admin’s interface.
Postconditions	Message will be shown to system admin which says “Disallow process is successful.”

Table 22: Disallow member

3.3 Usability Requirements

req42: A user shall use all related functions of the system wherever Internet connection is available.

req43: In case of a long conversation, user chat interface shall show little portion of latest conversation and user shall see remaining portion and older conversation by scrolling up.

req44: System admin shall be able to reach any related function at most 5 buttons.

req45: Vote and message buttons in worker chat interface shall be in bigger fonts and lively colours and worker messages shall be differentiated from user messages by their indentation and colours.

req46: A message search field shall be provided for both user and worker chat interfaces.

req47: System admin and IT staff shall have “back to top” button for message logs and error logs, respectively. This button shall return them to top of the logs.

3.4 Performance Requirements

req48: The system shall be available up to 900 users simultaneously.

req49: Message retrieval latency of both worker and user shall not exceed 600 ms.

req50: Backend of the system shall have at least 35 Gbps bandwidth and symmetric 1 Gbps upload/download speed.

req51: Server of the system shall support parallel computing and its hardware specifications shall be selected accordingly.

req52: Database of the system shall be synchronized with three identical but separate databases and these three databases shall work concurrently. They shall be designed as at least one of them work at any time instant (including maintenance times).

3.5 Logical Database Requirements

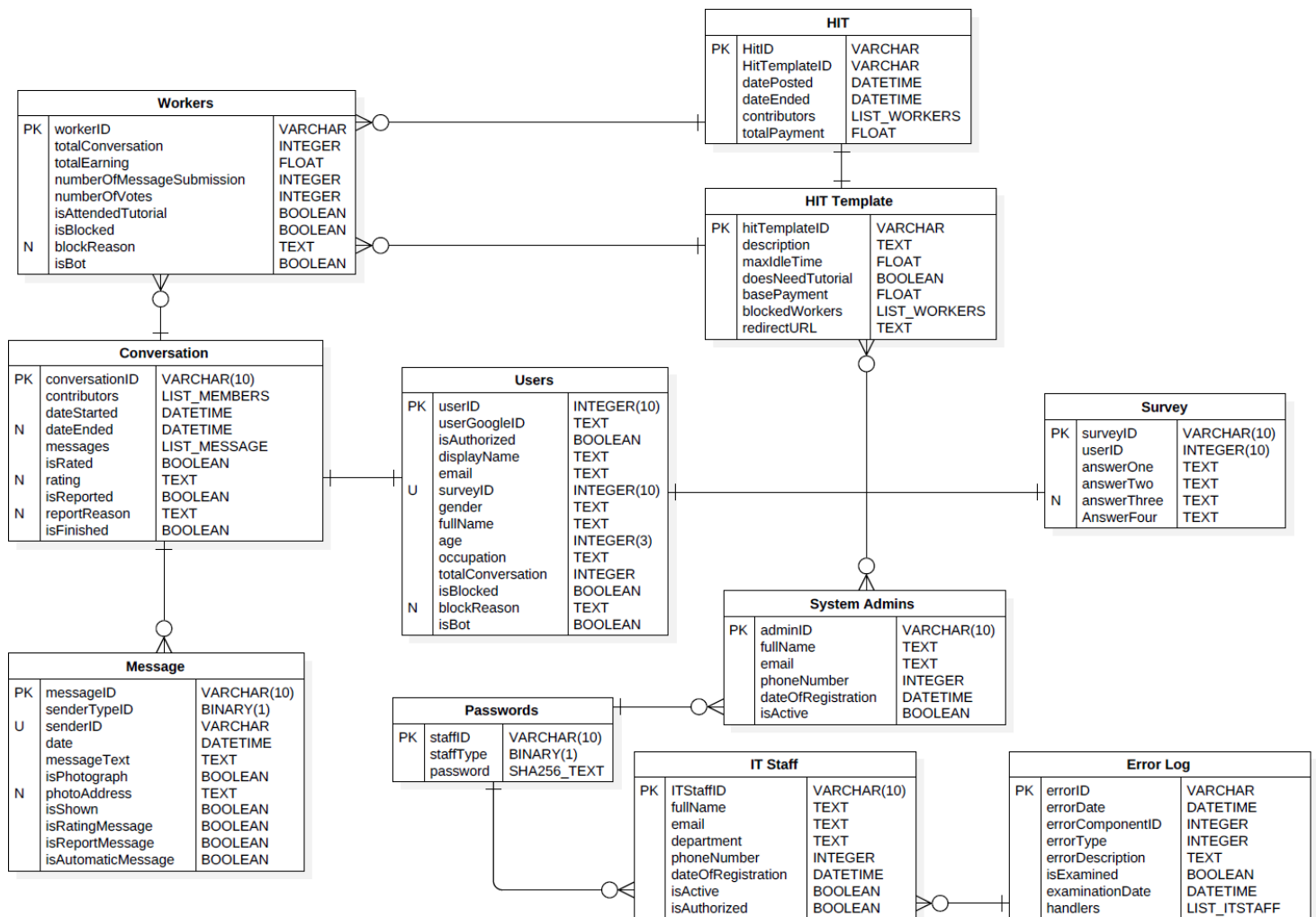


Figure 10: Logical Database Diagram

req53: Each conversation shall be started and processed when user writes first message after a conversation. Then, first message of conversation shall be written as "Conversation started" in database. User's first message shall come after that.

req54: A user entry shall be created when a user sign ups through registration interface.

req55: Passwords of staff shall be encrypted upon registration.

req56: Only system admins shall filter and search through Conversation and Message table.

req57: Only IT staff shall access Error Log table and search through it.

req58: Entries to Error Log table shall be added by Backend Error Management Interface when one or many problems are encountered during system runtime.

req59: Only system admins shall access to HIT Template table, however this access shall be only read-only. System shall automatically create and destroy templates restricted with designated constraints.

req60: Message entry shall be weak entry since a Message cannot exist without a Conversation has been started by user.

req61: HIT table shall be only internally accessible by system and it shall be used in absence or excess of workers.

req62: An entry to IT staff shall be made through one system admin's Management Interface. His/her entry in Passwords table shall be created with blank post. During his/her first login, staff shall be asked for password change and this password shall be written in Passwords table.

req63: Survey table entry shall be created when registration of user is complete.

req64: A System Admin table entry shall be created and removed by unanimous decision of current system admins. This decision shall be opened to vote in each system admins' Management Interface.

req65: An entry to Workers table shall be created when a new worker contributes to the system through Amazon MTurk.

req66: Only system admins can filter and search through Workers and Users tables.

req67: Integrity of database environment should be checked by Database Management Interface and IT staff. Database Management Interface shall check system twice in hour and IT staff shall run checks in maintenance hours.

req68: Database backup shall be done twice in a day.

3.6 Design Constraints

req69: The survey and registration data shall be stored for legal purposes.

req70: The system shall be designed in accordance with law of privacy.

req71: Name and Google ID of user shall be kept as classified.

3.7 Software System Attributes

3.7.1 Reliability

req72: Probability of data corruption of the system shall not exceed 0.015. This shall be acquired by three concurrent databases, two daily backups, Database Management and Backend Error Management Interface.

req73: Mean time of failure shall be one in 18 days.

req74: System shall be tested for reliability issues if one or more components are integrated into the system and before a new patch is released.

3.7.2 Availability

req75: System shall be available always excluding the maintenance time. Maintenance time shall depends on IT staff decision, however IT shall be done between 2 AM and 6 AM (GMT -4) and amount of maintenance time shall not exceed one and a half hour.

req76: All systems shall be restarted in at most 5 minutes after failure.

req77: IT staff shall be responsible for rapid response after critical failures. System shall be able to tolerate minor failures and keep main operations online.

req78: Total time when system is shutdown state due to failures shall not exceed 8 hours in one year excluding the maintenance hours.

3.7.3 Security

req79: Database Management Interface shall be durable to any database injection and man-in-the-middle attacks.

req80: Password column of Passwords table shall be encrypted using SHA256 and keys shall be changed from once a day.

req81: Remote access to system's development environment shall be read-only and suggested changes must be approved by a developer who access to servers with a physical connection.

req82: Authorization of users and workers shall be done through Google Hangouts and Amazon MTurk, respectively.

3.7.4 Maintainability

req83: Documentation of each development or bug fixing process shall be done in parallel with the procedure and international norms shall be followed.

req84: System shall combine COTS approach with modular design in order to increase maintainability of the system.

3.7.5 Portability

req85: System shall provide both desktop and mobile solutions for users and workers. Mobile solutions shall be provided by iOS and Android application of Google Hangouts for users and a mobile web site shall be developed for workers.

req86: System shall use multi-platform supported programming languages to ease migration of servers in failures.

req87: System shall mostly use libraries that is developed for at least two programming languages.

3.8 Supporting Information

Chorus Bot is a system that uses the power of crowdsourcing to provide an online personal assistance experience. This project is an initial step to automate conversation between system and users using machine learning frameworks.

Accounts of users and workers are fully controlled by Google Hangouts and Amazon MTurk, respectively. Any personal information of users and workers are not stored in the system unless their consent are taken.

4. Verification

5. Appendices

5.1 Assumptions and dependencies

5.2 Acronyms and abbreviations