SOFTWARE DESIGN DESCRIPTION

For Chorus Project

Bilal Özlü 1942614

Ali Şimşek 2099752

Version 1.0 (14.04.2018)

Table of Contents

List of Figures

List of Tables

1.Introduction

1.1. Purpose of the System

1.2. Scope

1.3. Stakeholders and their concerns

2. References

- IEEE Standard for Information Technology—Systems Design—Software Design Descriptions https://standards.ieee.org/findstds/standard/1016-2009.html
- Chorus: A Crowd-Powered Conversational Assistant by Lasecki https://dl.acm.org/citation.cfm?id=2502057
- Website of Chorus https://talkingtothecrowd.org/
- Advantages of HTML5 https://gotechark.com/blog/advantages-html5/

3. Glossary

4. Architectural Views

4.1. Context View

This part of the document explains Chorus system's interactions with user. (System components and the relationship between the components).

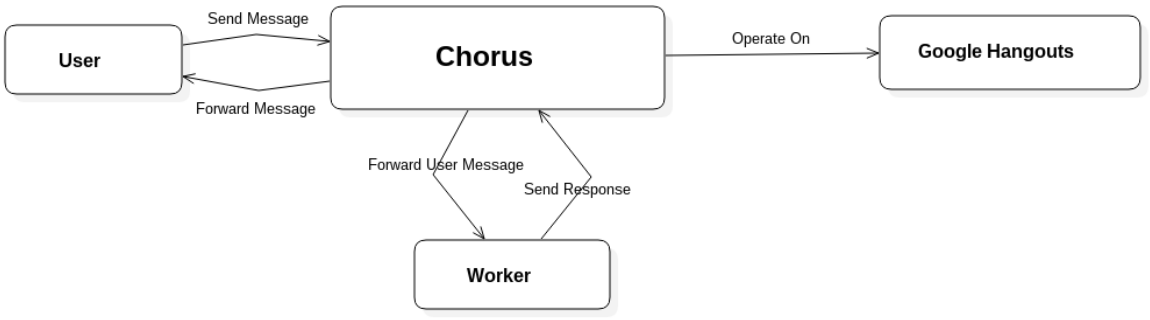

Figure 4.1.1: Context Diagram for Chorus System

Figure 4.1.2: Use Case Diagram for Chorus System

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

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

Table 4.1.1: Use-Case Descriptions Table

4.2. Composition View

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

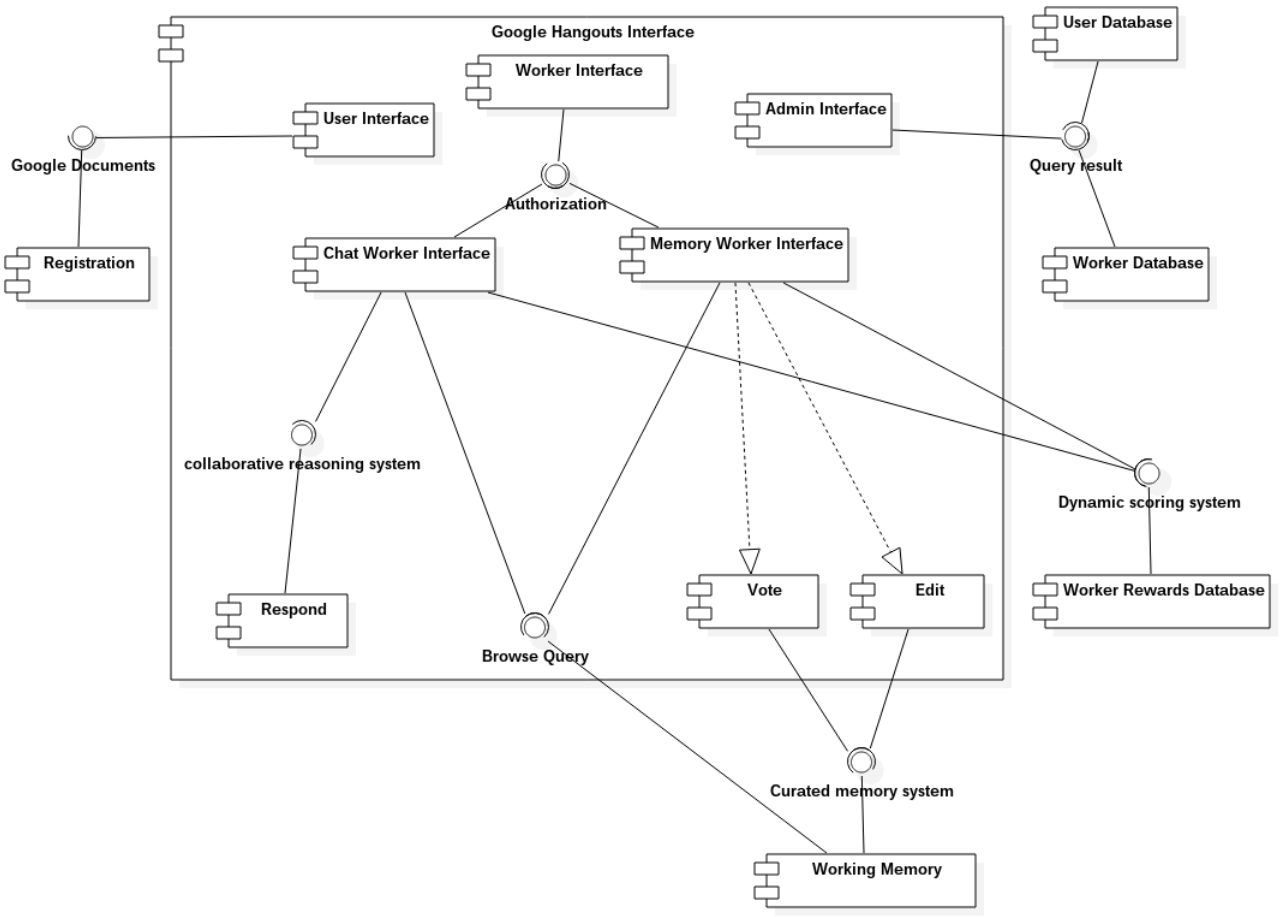

Figure 4.2.1: Component Diagram of Chorus

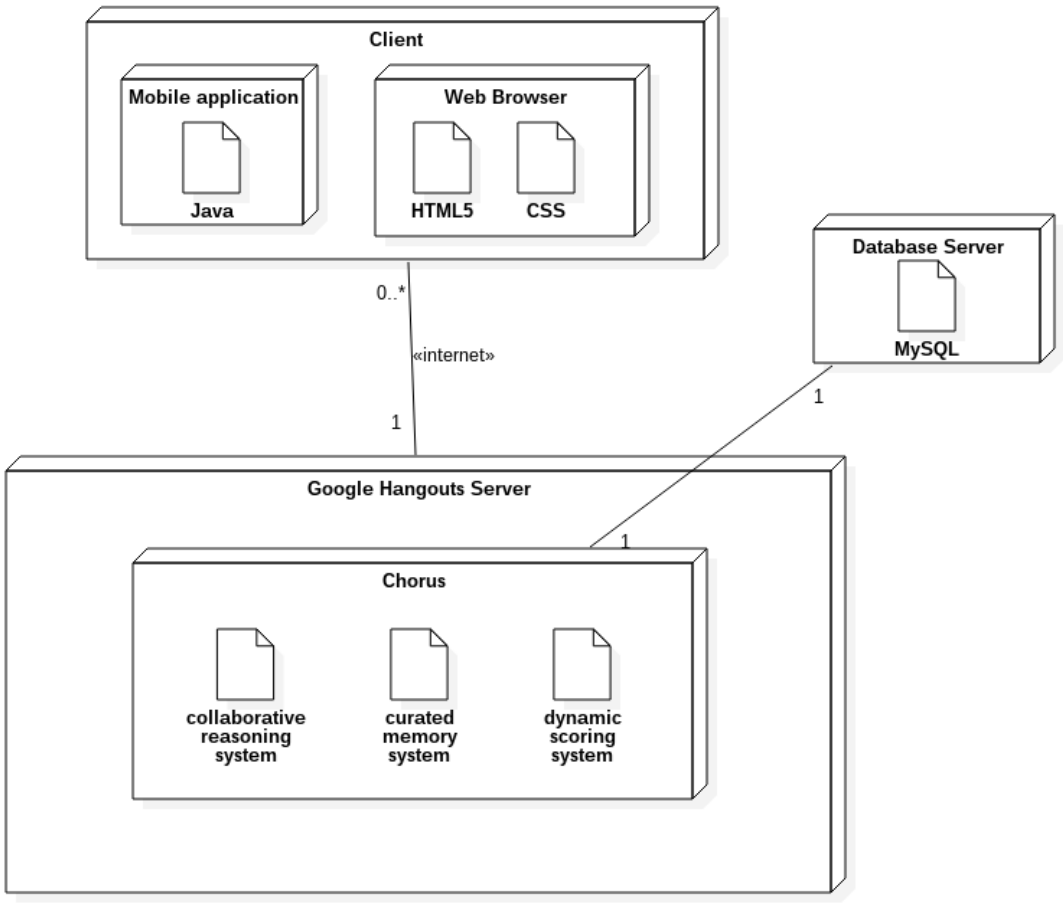

Figure 4.2.2: Deployment Diagram of Chorus

4.2.1. Design Rationale

- We used Java for mobile application to make it more portable and more secure. JVM provides extra security layer for the application and it also enables us to use the same code on different systems which makes application more portable.
- Our system use MySQL as database system since it is open-source and widely used.
- We used CSS on our browser application because its easy to maintain.By making one change to the website's CSS, elements in all the web pages will be updated automatically
- We used HTML5 because it enables designers to use cleaner and understandble code. Div tags can be replaced with some HTML5 elements.

- To make system more reliable, we seperated each of the databases. (User Database, Worker Database etc.). Even if one database encounters with an error, others will work.

4.3. Information View

4.4. Interface View