**Team Name: KryptoNight**

**Project Name: GoHelp**

**Software Design Document**

**Team Members**: A V S ADITYAVARDHAN, YADDANAPUDI VINAY, DHANUSH R H, KANDULA VENKATESWAR REDDY, PEDADA VENKATA HARSHA

**Revision**: 1.0

**Date**: 02-11-2020

# TABLE OF CONTENTS

# 1. Introduction

## 1.1 Purpose

This is the Software Design Document (SDD Version 1.0) for GoHelp, A Customer Worker Interaction platform. This document is intended to communicate the architecture and system design of GoHelp platform to the readers. SDD describes the detailed structure of the components of the GoHelp platform and the precise implementation details required to satisfy the requirements as specified in the Software Requirements Specification (SRS).

## 1.2 Scope

This Software Design Description (SDD) explains the structure of the components of the GoHelp customer worker interaction platform and some precise implementation details required to satisfy the requirements as specified in the Software Requirements Specification (SRS).

GoHelp software is mainly introduced to maintain a chain of workers residing in Rural and Semi Urban Areas in India. These workers are efficient in providing quality work but they don't have a right hands to join. They usually work under contractors and get less working wages.

The main agenda of this platform is to eliminate the middlemen contractors and make a communication between the customer and worker directly. GoHelp software helps users in electing best worker to work for relatively lower amounts. Many skilled workers can interact with new customers and can improve their working abilities earning more income than usual. This helps us to maintain a chain of workers, providing job opportunity with a fixed wages per hour work. This platform is online web application can be accessed all over the world.

## 1.3 Overview

This document gives the design related detailed information of the GoHelp Customer Worker Interaction Platform. The system provides the information about System Architecture, Data Design, Component Design, Human Interface Design and the Requirements Matrix of the GoHelp Platform

## 1.4 References

**Name of the author**: Mr. Serguei Mokhov

**version**: 1.0

**Link**: https://arxiv.org/ftp/arxiv/papers/1005/1005.0595.pdf

## 1.5 Definitions and Acronyms

1. Interaction Platform – It is a communication method adopted in the platform where worker and customer interact in the chat window.
2. Catalogue – It is a representation of data into rows and columns containing filters to select the required data.
3. Filter – It is a condition which helps to remove the data which fails to obey the condition.

# 2. System Overview

GoHelp is a Customer Worker interaction platform introducing workers to interact directly with the customers. This platform brings the new working style of the workers invoking a direct communication between the worker and the customers eliminating the involvement of third parties like contractors, mediators, brokers.

Both Workers and Customers can register into the platform. Workers can update their skills which includes their past experience, their working skills. Our platform will collect skills from the database and display to the customer. Customer will select appropriate filters provided and contact the worker with the chat provided.

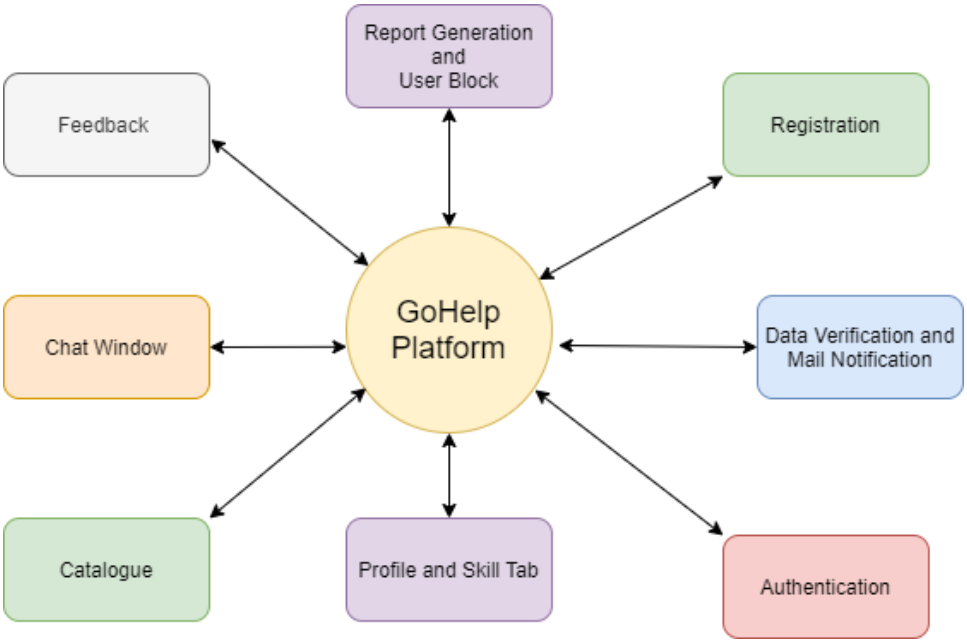
Once customer and worker interact in the chat window, customer will allow worker to complete the work by updating the work acceptance. Worker and customer can provide feedback for the work they reported. The system then updates the rating of worker with the feedback.

Customer and Worker can block the other worker / customer if they are not subjected to the terms and conditions mentioned in the site. Admin can block the user / customer to access the platform and can generate reports on the number of workers / customers utilized, money gained...,

Customers / Workers will periodically receive notifications from the application when a new customer / worker comes to interaction, customer accept the application, report statements mail.

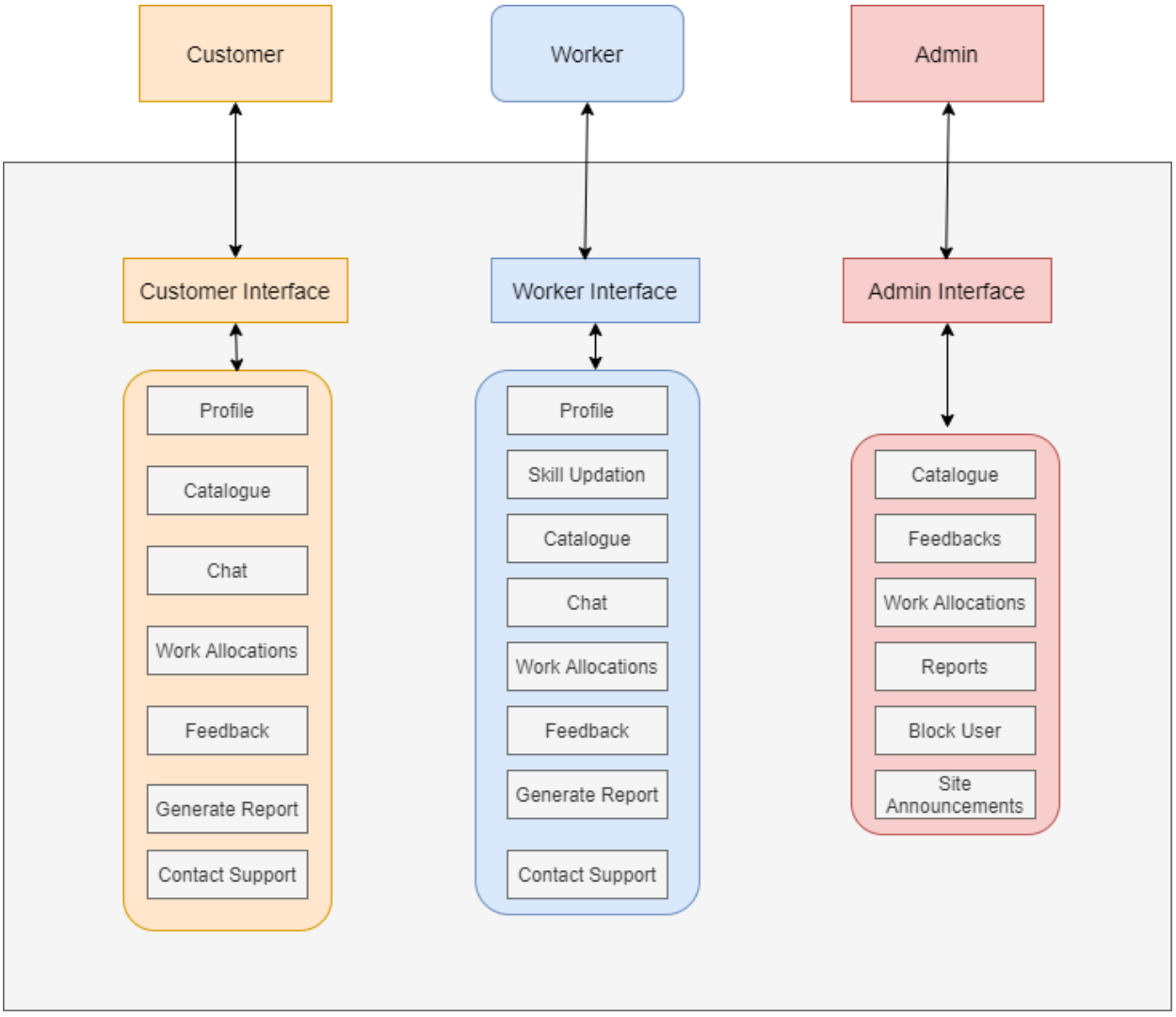
# 3. System Architecture

## 3.1 Architectural Design




GoHelp platform is a web based interactive platform which involves user registration, verification and notification, authentication, profile and skill creation, catalogue, chat window, feedback and report generation components. These components integrate together and provides the best-in-class worker-customer interaction platform.

In this platform we have three different users' workers, customers and an administrator respecitvely. A broad functionality of these three users are given below




### 3.1.1 Registration

**Description**: Registration component takes care of user registration. Users can be worker, customer. Admin registration will be done internally.

**Dependency**: Independent Component

### 3.1.2 Data Verification and Mail Notification

**Description:** Data verification will take care of verifying, validating the user data entered in the application. It also delivers email and text notifications to the users.

**Dependency:** Independent Component

### 3.1.3 Authentication

**Description:** This component helps users to login (authenticate) into the platform.

**Dependency:** Registration, Data Verification and Mail Notification

### 3.1.4 Profile and Skill Tab

**Description:** This component helps to create and update, user profile and worker skills in the platform.

**Dependency:** Authentication.

### 3.1.5 Catalogue

**Description:** This component will make a catalogue with the skills provided by the worker so that customer can check the worker availability, cost, ratings. This helps customer to elect the right worker for his work

**Dependency:** Profile and Skill Creation, Authentication

### 3.1.6 Chat Window

**Description:** This component provides a chat interface where a customer and worker can interact and fix an appointment to complete the work.

**Dependency:** Profile and Skill Creation, Authentication, Catalogue

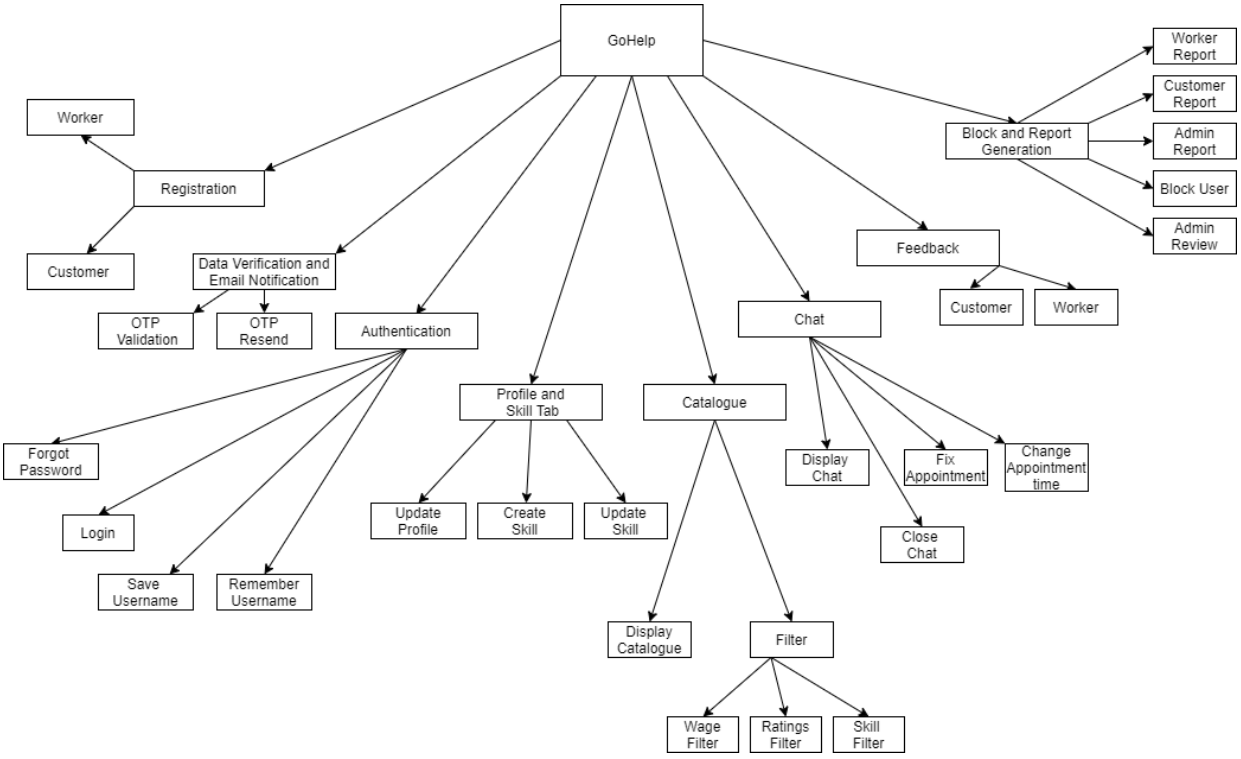
### 3.1.7 Feedback and User Block

**Description:** This component takes user (both customer and worker) feedback in the system. These feedbacks helps other customers / workers to select the best one next time.

**Dependency:** Authentication, Profile and Skill Creation

## 3.2 Decomposition Description

Each of these components are decomposed into sub-components / sub-systems. Each of these sub-components inter related to each other to perform the entire component functionality.




Each of the high level component is sub divided into smaller components / subsystems that perform functionalities.

**Registration Subsystem:**

**Worker**: It takes care of worker registration

**Customer**: It takes care of worker registration

**Data Verification and Email Notification:**

**OTP Validation:** OTP Entered by the user will be validated in the system

**OTP Resend:** OTP can be reshared with the user

**Authentication:**

**Forgot Password:** User can change his password incase if he forgets

**Login:** Authenticates user into the system

**Save Username:** Username will be stored in the system

**Remember Username:** Remembers the username of the user by setting username as cookie

**Profile and Skill Tab:**

**Update Profile:** User can update the profile mentioned at the time of registration

**Create Skill:** Worker can create his skill set by mentioning his domain knowledge, experience

**Update Skill:** Worker can update the skill set created in the platform

**Catalogue:**

**Display Catalogue:** Displays the worker skill set as a catalogue

**Filters:** Use filters to remove the unnecessary data.

**A) Skill Filter:** Update the worker data based on skill

**B) Wage Filter:** Update the worker data based on wage

**A) Rating Filter:** Update the worker data based on the rating

**Chat:**

**Display Chat**: Displays the user chat

**Close Chat**: Closes the chat so that user cannot proceed

**Fix Appointment**: Fixes the worker user appointment

**Change Appointment Time**: Appointment time of worker can be changed

**Feedback**

**Worker:** Worker can provide feedback to the customer

**Customer:** Customer can provide feedback to worker

**Block and Report Generation:**

**Block:** User can block another user by requesting the admin

**Worker Report**: Worker Activity is mentioned as a report

**Customer Report**: Customer Activity is mentioned as a report

**Admin Report**: Customer and Worker Activity are mentioned as a report

## 3.3 Design Rationale

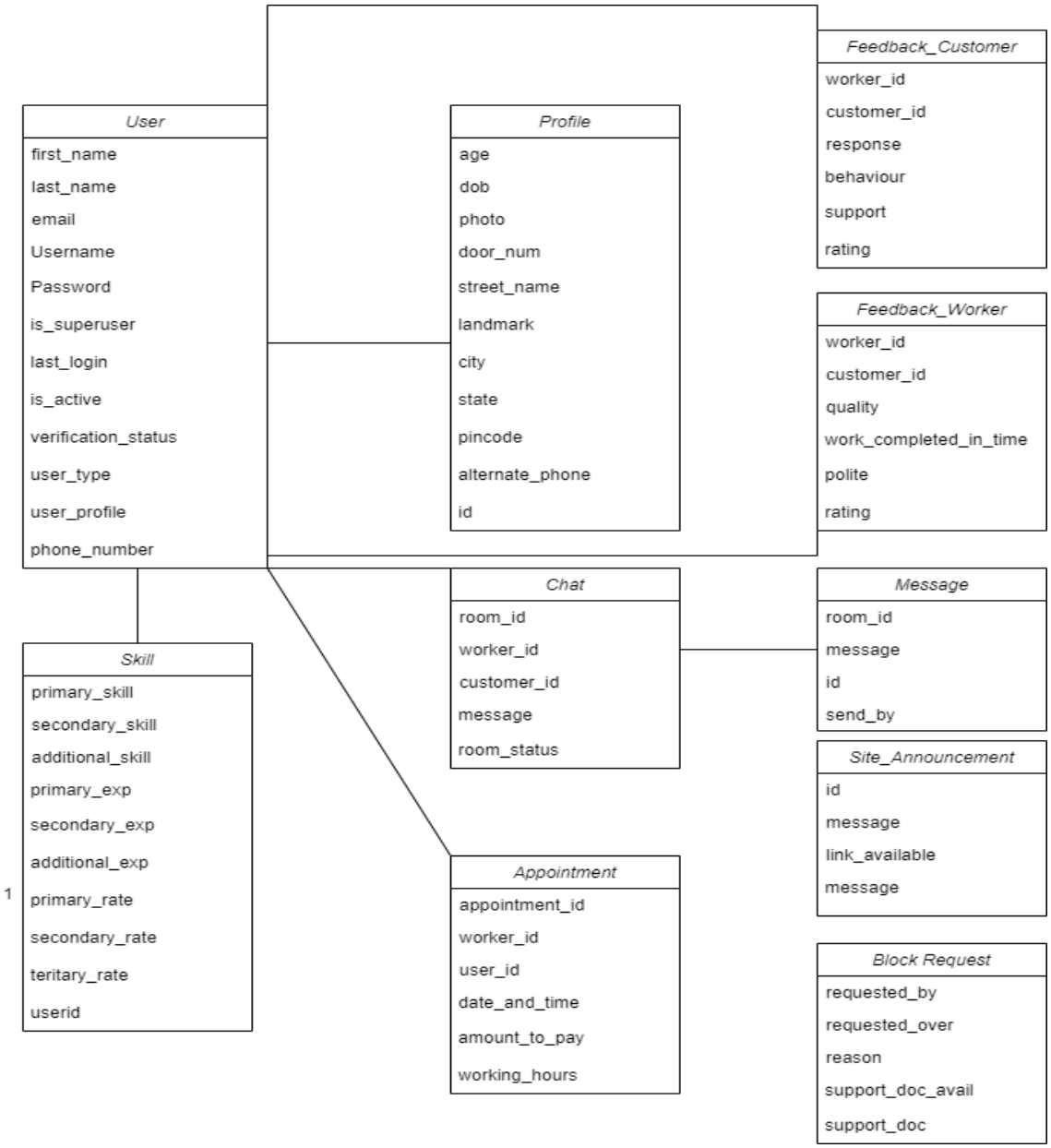
The major advantage of such design is reducing the latency time required to perform an operation in the platform. This GoHelp Worker Customer Interactive platform can be accessed anywhere / anytime with unlimited number of users. We came up with this design to provide the hassle-free experience to both customers and workers.

Another major advantage of this design is load handling. Server can handle load easier if we can reduce the time taken to complete the operation. All the high-level components are made to smaller components so that server can dedicatedly run each smaller component easier than running a huge component involving multiple connections. This design principle avoids deadlocks and maintain each component as independent as possible.

# 4. Data Design

## 4.1 Data Description

GoHelp Worker Customer Interaction Platform data is designed as follows.




We have adopted database normalization in order to reduce the data redundancy and improve data integrity in the application. These database tables contain foreign key constraints to execute joint queries in the application

## 4.2 Data Description

The below table provides an overview of the database columns below.

| Table Name | Column Name | Description | Datatype | Size | Constraint | Null values allowed |
|---|---|---|---|---|---|---|
| User | first_name | First Name of the user | varchar | 50 | - | no |
| | last_name | Last Name of the user | varchar | 50 | - | yes |
| | email | Email Address | varchar | 50 | - | no |
| | username | Username of the user account | varchar | 50 | Primary key | no |
| | password | Password for the account | varchar | 50 | - | no |
| | is_superuser | Checks whether user is admin | boolean | - | - | no |
| | last_login | User last access to site | date | - | - | no |
| | is_active | Determines whether user can access account | boolean | - | - | no |
| | verification_status | Determines whether user credentials verified | boolean | - | - | no |
| | user_type | Identify the user account type (Worker / Customer) | varchar | 30 | - | no |
| | user_profile | Profile id for the user | int | 10 | Foreign key | yes |
| | phone_number | Mobile number of user | int | 10 | - | - |
| Profile | id | Identity for the profile record | int | 100 | Primary Key | no |
| | age | Age of user | int | 100 | - | no |
| | dob | Date of Birth | date | - | - | yes |
| | photo | User profile photo | image | - | - | yes |
| | door_num | Door Number of user | varchar | 100 | - | no |
| | street_name | Street Name of the user address | varchar | 100 | - | no |

| | | | | | | |
|---|---|---|---|---|---|---|
| | landmark | Nearby location of the user address | varchar | 100 | - | no |
| | city | City at which user resides | varchar | 100 | - | no |
| | state | State at which user resides | varchar | 100 | - | no |
| | pincode | Pincode of location which user resides | int | 6 | - | no |
| | alternate_phone | Alternate mobile number of user | int | 10 | - | yes |
| Skill | primary_skill | Primary skill of the worker | varchar | 100 | - | no |
| | secondary_skill | Secondary skill of the worker | varchar | 100 | - | yes |
| | additional_skill | Additional skill of the worker | varchar | 100 | - | yes |
| | primary_exp | Experience in Primary skill | int | 100 | - | no |
| | secondary_exp | Experience in Primary skill | int | 100 | - | yes |
| | additional_exp | Experience in Primary skill | int | 100 | - | yes |
| | primary_rate | Wage to pay for primary skill per hour | int | 100 | - | no |
| | secondary_rate | Wage to pay for primary skill per hour | int | 100 | - | yes |
| | additional_rate | Wage to pay for primary skill per hour | int | 100 | - | yes |
| | userid | Username of the user | varchar | 50 | Foreign Key | no |
| Chat | room_id | Window id for the chat | int | 100 | Primary Key | no |
| | worker_id | Username of worker | varchar | 50 | Foreign Key | no |
| | customer_id | Username of customer | varchar | 50 | Foreign Key | no |
| | message | ID of Messages sent in the chat | int | 100 | Foreign Key | no |

| | | | | | | |
|---|---|---|---|---|---|---|
| | room_status | It represents whether chat can continue or not | boolean | - | - | no |
| Message | id | Identifier for the message | int | 100 | Primary Key | no |
| | room_id | Identifier for the chat window | int | 100 | Foreign Key | no |
| | message | Message sent | varchar | 1000 | - | no |
| | send_by | Username of the user | varchar | 50 | Foreign Key | no |
| Appointment | appointment_id | Identifier for worker customer appointment | int | 100 | Primary Key | no |
| | worker_id | Username of worker | varchar | 50 | Foreign Key | no |
| | customer_id | Username of customer | varchar | 50 | Foreign Key | no |
| | date_and_time | Time and Date at which appointment was confirmed | datetime | - | - | no |
| | amount_to_pay | Wage fixed for work | int | 100 | - | no |
| | working_hours | Number of hours in which worker was to complete the work | Int | 100 | - | yes |
| Feedback_Worker | worker_id | Username of worker | varchar | 50 | Foreign Key | no |
| | customer_id | Username of customer | varchar | 50 | Foreign Key | no |
| | quality | Work quality | int | 2 | - | no |
| | work_completed_in_time | Whether worker followed dead lines | boolean | - | - | no |
| | polite | Worker behaviour | boolean | - | - | no |
| | rating | Overall Rating for the worker | int | 10 | - | no |
| Feedback_Customr | worker_id | Username of worker | varchar | 50 | Foreign Key | no |
| | customer_id | Username of customer | varchar | 50 | Foreign Key | no |

| | | | | | | |
|---|---|---|---|---|---|---|
| | behaviour | Customer Behaviour | varchar | 50 | - | no |
| | support | Customer Support | varchar | 50 | - | no |
| | response | Customer responding nature | varchar | 50 | - | no |
| | rating | Overall Rating for the worker | int | 10 | - | no |
| Site_Announcements | id | Identifier of announcement | int | 10 | Primary key | no |
| | message | Message to Scroll | varchar | 200 | - | no |
| | link_available | Used to know whether a “click here to display” should scroll or not | boolean | - | - | no |
| | link | Reference url | varchar | 200 | - | yes |
| Block Request | requested_by | Username of person request to block other account | varchar | 50 | Foreign Key | no |
| | requested_over | Username of person on which action has to be taken | varchar | 50 | Foreign Key | no |
| | reason | Reason to Block | varchar | 500 | - | no |
| | support_doc_avail | Any supporting statement available | varchar | 3 | - | no |
| | support_doc | Proof to block | image | - | - | yes |
| | | | | | | |

# 5 Component Design

GoHelp Worker Customer Interaction platform has many sub components. Each of the component work is discussed below

## 5.1.1 Registration

This component takes care of registration of users in the gohelp platform. This component will receive user information, process it and sends the information to database. Some of the sub components involved are

**Worker Registration**

Worker is a user who has the ability to perform the work. He can perform the work if provided by the user. The worker is allowed himself to register into the application. Worker has to enter his first name, last name, age, date of birth, address, proof of residence, email address, phone number, alternate phone number, username, password, photograph. He will be provided with the check box stating the terms and conditions to accept to make the registration.

Once details are successfully submitted by the worker, a validation will be made before pushing the data to the database. Email, Mobile Numbers will be checked using One Time Password delivered to the user. Once these details are verified successfully a record is created in the database table where the details entered by the worker are stored.

If the details are successfully stored, "Registration Success" is displayed on the screen. Otherwise "Registration Failed, Try again Later" message will be displayed.

**Customer Registration:**

Customer is a user who needs a worker to perform the work. He has some work to be done if a worker is available. The Customer is allowed himself(herself) to register into the application. Customer has to enter his first name, last name, age, date of birth, address, proof of residence, email address, phone number, alternate phone number, username, password, photograph. He will be provided with the check box stating the terms and conditions to accept to make the registration.

Once details are successfully submitted by the customer, a validation will be made before pushing the data to the database. Email, Mobile Numbers will be checked using One Time Password delivered to the user. Once these details are verified successfully a record is created in the database table where the details entered by the customer are stored.

## 5.1.2 Email Verification and Email Notification

In this component the activities related to email verification, email notification delivery will be done. There are some sub sytems involved in this component.

**OTP Validation:**

This system will send a One Time Password (OTP) to the user. This component can be reused by providing the OTP, email or mobile number, message to which the information must be delivered should be provided as inputs.

Once the information received by the system, we first address whether the information to be sent to mail or phone number. If it is email, the send_mail() function will be initiated where the following parameters are provided.

- from_email : Email from which the mail has to be sent
- to_email : Email to which the mail has to be sent
- message : Message to send
- Subject : Email Subject

Once the information is set, a request will be placed to the Simple Mail Transfer Protocol (SMTP) server managed by google. The google server will deliver that email to the recipient (to_email) from the from email with the body as message

User will be shown a form where the OTP shared to his email must be entered. The OTP entered by the user will be cross checked with the OTP delivered by the system. In that case if OTP matches, the user email is successfully verified and a message “Verification Success” is displayed else “Verification Failed” will be displayed.

OTP Validation for phone number will be a bit different than the mail id. Instead of sending request to SMTP server, we send the same request to the twilio server. Twilio is a third party service used to deliver the messages. User will be shown a form where the OTP shared to his mobile number will be entered.

The OTP entered by the user will be cross checked with the OTP delivered by the system. In that case if OTP matches, the user email is successfully verified and a message “Verification Success” is displayed else “Verification Failed” will be displayed.

**OTP Resend:**

The One Time Password can be resended to the user if user requested to do. Once the user clicks resend button, the mail / mobile number which user previously entered will be shared with a new OTP and the system will be updated with the new value.

The steps mentioned in the OTP Validation will be used to validate the phone number / email id.

**Email Notification:**

This component send email notifications to the users. It will be done by inputting the email id, message to which the information must be delivered.

Once the information received by the system the send_mail() function will be initiated where the following parameters are provided.

- from_email : Email from which the mail has to be sent
- to_email : Email to which the mail has to be sent
- message : Message to send
- Subject : Email Subject

Once the information is set, a request will be placed to the Simple Mail Transfer Protocol (SMTP) server managed by google. The google server will deliver that email to the recipient (to_email) from the from email with the body as message.

## 5.1.3 Authentication

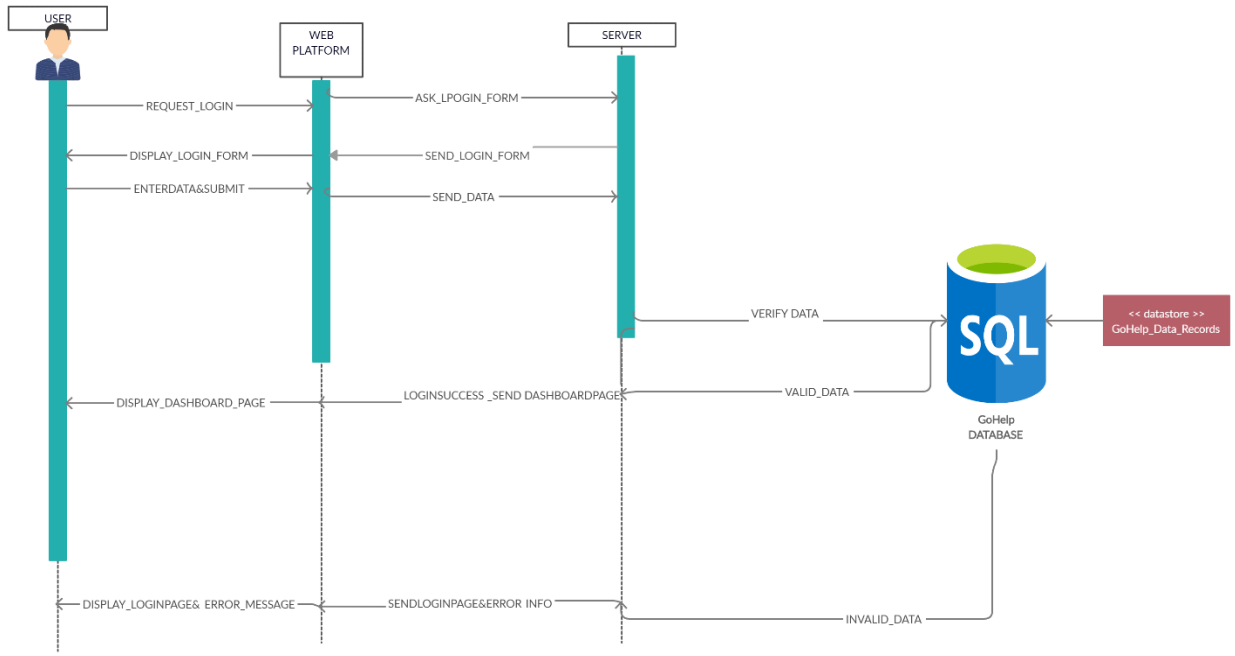
User Details are authenticated (check whether the information entered is valid or not) before accessing the application. The sub-systems in this component are as follows

**Login**

User login will be performed in the system where a user is provided with a form to enter the username and password set at the time of registration. Once the user enters the details and clicks the submit button, the details will be checked in the database. Each record in the database will be matched to know whether username and password exist for any record.

If exist the user will be marked as Logged In and update the last login record with the present time and date. If user details are not matched we display “Username / Password” not matching to the user and request him to re-enter the username and password. If he enters wrong credentials, a small button named “Forgot Password” will be displayed every time to change his password.

The sequence of actions involved in the login component are as follows.




**Forgot Password:**

This component is used to change the user password set to his account. On clicking this button, user will be requested to enter his email id to send the One Time Password. Here, the main objective of user email is to verify whether any user registered with this email Id or not. The email id will be unique in the application as it will be checked before user allowed to register in the application to avoid confusion in email notification and delivery.

Once a record with the email id is founded out, **OTP Validation** component will be executed. If OTP Validation is done successfully, user will be displayed a form to change his password. This form request user to enter his new password and ask to confirm the new password entered in the form.

Once user clicks the submit button, user password will be encrypted and saved in the database. An email notification will be sent to the user stating "Your Password has been changed successfully".

User can change his password at any time in his dashboard provided in the platform. If it was done there, the validation step is neglected as user is already logged into the system. Once changed successfully, user will be automatically logged out from the application.

**Remember username:**

User can request the system to remember his username so that he can enter the password directly. To perform this component, user must enable cookies in the web browser. If user is failed to enable, this component will return Failure messages.

This is a checkbox shown at the time of login. Once user login is done with the checkbox, we will set up a cookie in the browser so that user can directly enter the password from next time during login into the system

### 5.1.4 Profile and Skill Tab

User can update his profile entered at the time of registration. Worker can update the skill set in this platform. Some of the components are mentioned below.

**Update profile**

User can update his profile by filling the form which contains address, alternate phone number. the address includes door number, street name, locality, state of residence. Once user clicks the submit button, the user details will be updated with the latest data send by the user and shows "Updated Successfully" message if the records are saved. Else "Try again later" will be displayed on the screen.

**Create Skill**

Create skill component helps worker to create the skill set for getting work opportunities. To create a skill set, worker must enter his primary skill, secondary skill and alternate skill in which he can work on. He needs to enter the experience of working on primary, secondary and alternate skill. He need to enter the wage needed per hour for working on the skill other than the labour charge.

Once these details are entered successfully, we will create a new record in the database with the data provided by the worker. If the data is successfully saved, we display "Skill set created successfully" to the worker. Otherwise, we display "Try again" message so that worker can try to update his skills set again.

**Udpate Skill:**

Worker can update his skills which are entered earlier in the application. Once the worker clicks on Update Skill, the skill data which was previously created by that worker will be collected and displayed to him. Worker can edit that data by providing the latest information. Once editing is done, he can click the submit button to update data

Once these details are received, we will search for the record in the database and updated with the data we received from worker in the database If the data is successfully saved, we display "Skill set updated successfully" to the worker. Otherwise, we display "Try again" message so that worker can try to update his skills set again.

## 5.1.5 Catalogue

Worker skill set present in the database and set it in a catalogue where a customer can see the different kinds of workers present in the application. There are some sub systems involved in the component.

**Display Catalogue**

A catalogue will be displayed to the customer. All the skillset information will be collected from the database. These skill set will be arranged as a card where each card will represent the information of the worker. This information will be the primary skill set of workers, secondary skill set of workers, additional skill set of workers, along with the workers' photo and the rating provided by the other users in this platform.

Newly joined workers will have a rating zero and displayed newly joined. These cards are arranged as primary skills first followed by secondary skill and additional skills respectively. Once the customer believes that this worker can suit to the work I need, he can click the "Start Chat" button to start.

When the customer clicks the start chart button, we will first check whether the user has already in chat with the worker or not. If he is already opened a chat we will redirect to the chat board provided in the application else a new chat will be created.

**Filters:**

Filters are the essential components involved in any application. They always help us to get the correct data removing the unnecessary ones in the application. GoHelp catalogue component also provide some filters to the customer to select the workers.

**Skill Filter:**

Customer can use the skill filter if he wants to select the worker based on the skill. This skill filter contains all the skills (primary, secondary and additional skills) that our platform holds. Once the customer selects the skill and click the submit button, the request will be sent to the server.

Server will fetch the records from the skill set again from the database and displays it to the customer. The customer can re-apply the filter if he wishes to do so. There is no implication on the number of times the customer can use the filter. The customer can only apply one skill to filter at the time. The main reason is to provide chance to other workers as well in the platform.

**Wage Filter:**

Customer can use the wage filter if he wants to select the worker based on the charge to pay per hour. This wage filter contains the wages in range ($5-$10, $10-$20). Once the customer selects the wage and click the submit button, the request will be sent to the server.

Server will fetch the records from the skill set again from the database and displays it to the customer. The customer can re-apply the filter if he wishes to do so. There is no implication on the number of times the customer can use the filter. The customer can only apply one wage to filter at the time.

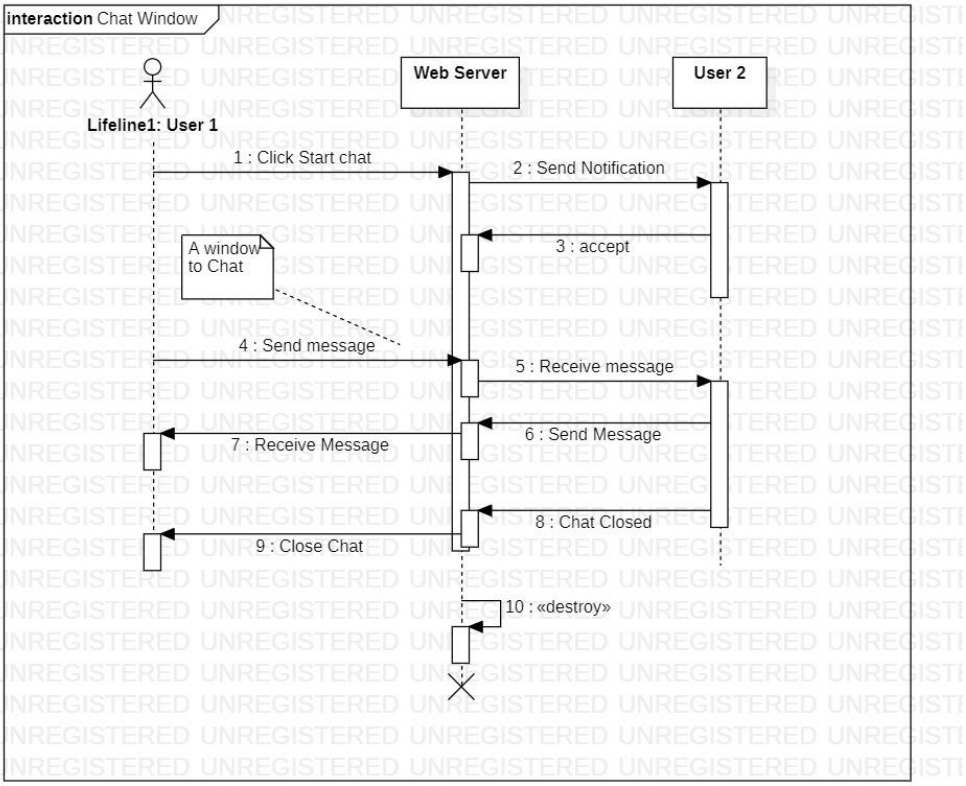
**Rating Filter:**

Customer can use the rating filter if he wants to select the worker based on the ratings provided by other customers to that worker in the platform. This rating filter contains the ratings 1,2,3,4,4.5,5. Once the customer selects the rating and click the submit button, the request will be sent to the server.

Server will fetch the records from the skill set again from the database and displays it to the customer. The customer can re-apply the filter if he wishes to do so. There is no implication on the number of times the customer can use the filter. The customer can only apply one rating filter at the time.

### 5.1.6 Chat Window

Customers and workers can engage in a chat directly so that both can discuss more on the work that customer expects the worker to do with the wage provided. User can engage in the sequence of steps provided in the application as follows.




**Display Chat:**

The user can engage in a conversation by clicking the chat section provided in users' dashboard. Once the user clicks on that chat name, the user will be displayed with the window where he can send the message to other user in the application.

User can enter the message that he wishes to communicate with the other user and can click the submit button. Once user clicks the submit button, the message will be sent to the server and stored in the database.

Once the other user comes to online, the new messages will be displayed and he can respond to the user back in the same application. The users can continue their conversation and discuss about the nature of work, work wage and availability.

**Close Chat:**

User can close the chat if he wishes to discontinue the conversation with the worker. The user will be provided with a button to close the chat and once clicked the request will be sent to the server.

Server will trigger an email id to both the users engaged in the conversation that "Your Chat window has been closed". The chat window status will be set to "Closed" in the database. The users can no longer continue the conversation in the application. User can reopen the chat by

requesting the admin in the contact support in the form provided in the dashboard. Admin can reopen the chat and user can again continue the conversation.

Admin can close the chat without the intervention of users if they are not abiding to the terms and conditions provided at the time of registration. If the user continues to engage in the same way the admin can close his account which disables the user to access the platform and the services provided in the platform.

**Fix Appointment:**

Customer can fix the appointment with the consent with the worker once they are engaged in the conversation. Customer needs to fill the working wage provided, working hours and date and time of appointment where the worker visits the customer house to do the work. Once the Customer clicks the submit button, the data is sent to server.

Server will create an appointment and displays the Success message once appointment is fixed. Worker can't initiate the appointment as customer only has access to the application. Both the users will receive the mail notification relating the appointment, along with the customer address, date and time of appointment.

**Change Appointment**

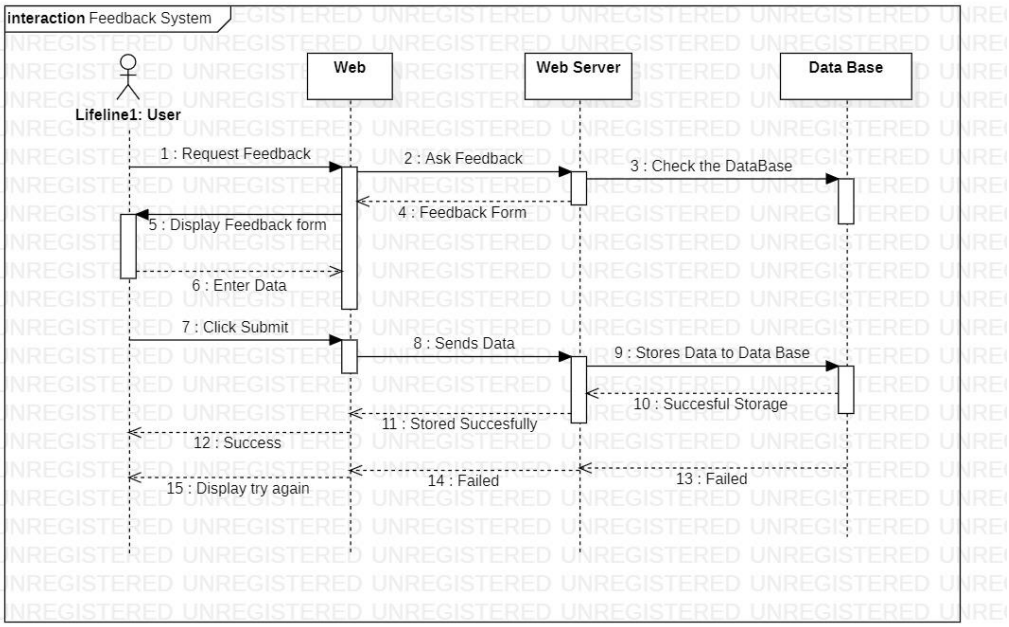
Customer and worker can change the appointment timings when the worker / customer are not available at the time of appointment which was fixed earlier. The appointment change will be done by entering the new timings and click on the submit button.

Once the new appointment timings are submitted, the server updates the data in the database and sends the email alert to both customer and worker stating "Change of Appointment".

Customer and worker can also request admin to do the change if they are not able to perform. They can contact support to do that.

## 5.1.7 Feedback

Worker and customer can provide feedback to each other for the work being done by the user. They can submit the feedback easily by clicking the link on the dashboard. A user can only provide one feedback at a time.




**Worker Feedback**

Worker can fill the feedback to the customer. He will be provided some parameters like customer behaviour, customer response, customer support and the rating for the customer. Once the submit button is clicked, the data will be sent to the server.

A new record of feedback will be created with the details provided by the worker and displays “Feedback Submitted Successfully” if the feedback is stored else “Feedback Submission Failed. Try again later” will be displayed.

This feedback will be shared to the co-workers when the chat is opened in the platform.

**Customer Feedback:**

Customer can fill the feedback to the worker for the service provided. He will be provided some parameters like work quality, work completed on time, worker politeness along with the rating. Customer once filling the form can click  customer. Once the submit button is clicked, the data will be sent to the server.

A new record of feedback will be created with the details provided by the worker and displays “Feedback Submitted Successfully” if the feedback is stored else “Feedback Submission Failed. Try again later” will be displayed.

This feedback will be shared to the co-workers when the chat is opened in the platform.

### 5.1.8 Block and Report Generation

The users of gohelp platform can request the platform to generate the reports for their usage. It can be done at anytime in the application. The report will be automatically downloaded from the platform.

**Worker Report**

Worker can download the report on clicking the tab in the dashboard. It includes the number of hours he worked, no of customers liked his work, along with the feedback provided by the customer.

It is an autogenerated report and will be downloaded in pdf format They can use this report if they wish to work on some other company and will be helpful in the time of recruiting. Report intakes the database tables feedback, appointments, catalogue data to provide the report.

**Customer Report**

Customer can also download the report on clicking the tab in the dashboard. It includes the no of workers he encouraged to perform the work, along with the feedbacks provided by the worker. It is an autogenerated report and will be downloaded in pdf format They can use this report if they wish to establish a community of workers under him.

Report intakes the database tables feedback, appointments in providing the report.

**Admin Report:**

Admin can also generate the overall report for budgeting in the company. He can download it from the dashboard. It includes overall information on number of users came into the platform, working hours, revenue generated by workers, along with the feedbacks provided by the workers.

Report intakes the database tables feedback, appointments in providing the report.

**Block User:**

Admin can block the user if any user is not obeying the terms and conditions set at the time of registration. Users can request the admin to block a user if they are not following the Terms and conditions. A worker can block a customer and vice versa only.

They can fill the support team with the username of user, document of proof to block and reason to block the user. Support team will take care of blocking the user. User will be notified if the admin blocked access to him.

These are the different sub-systems used in the application.

# 6 Human Interface Design

Human interface is the most important interface that attracts the users to the platform. GoHelp platform has a responsive user interface that can be accessed across all the smart phones, laptops and desktop computers. The only requirement is to have a proper internet connection and the web browser for the system.

## 6.1 Overview of the Interface

The users in this system are customers, workers and admin.

Worker can first click on the registration button and fill the registration form with the user type as worker to get reached to the customers. Once the registration is done, he can login in the platform with the username and password. Once login is successful, he will be displayed his own dashboard containing the profile updation, skill set generation..,

Worker can update his profile and generate a skill card to display to customers. He need to provide his work experience, skills he is aware of and confident to perform along with the wage (cost) to pay by customers per hour. Customers will come in touch with the worker and can engage in the conversation. They can discuss about the nature of work, working hours and wage provided per hour.

Customer can accept his application and he receives a email confirmation for the work along with the address. He can keep track of past works received in the platform by generating a report and can contact support if he wishes to communicate anything to the administration department. Worker can provide feedback to the customer to help the other workers in the platform.

Customer can first click on the registration button and fill the registration form with the user type as customer to get access to the workers. Once the registration is done, he can login in the platform with the username and password. Once login is successful, he will be displayed his own dashboard containing the profile updation and a catalogue of workers.

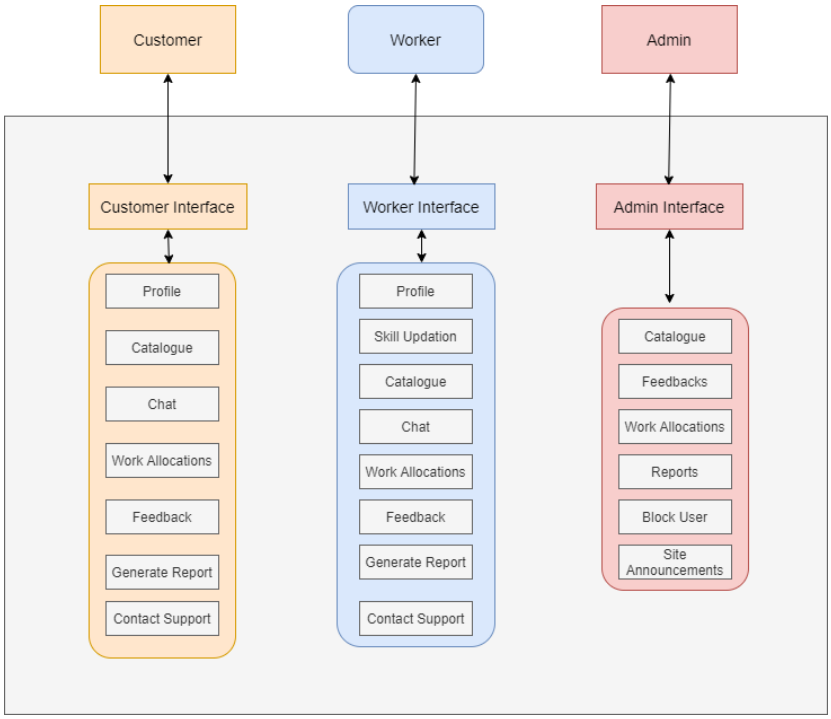
Each card in the catalogue displays the skill set (ability) of the worker to perform the job and can click to chat if he want to engross into the conversation. Both customer and worker can discuss about the nature of work, working hours and wage to pay per hour.

Customer can make an appointment and will receive an email confirmation for the work appointments. He can keep track of past works provided to workers in the platform by generating a report and can contact support if he wishes to communicate anything to the administration department.

Customer can provide feedback to the worker to help the other customers in the platform. Both customer and worker can alter the timings of appointment with a single click in the platform and receive the updated timings.

Admin can also able to access the dashboard and can perform all the activities that a user (customer) can work and can be able to block user if they are not obeying the terms and conditions. Admin can download a report of the platform usage with a single click on the dashboard.

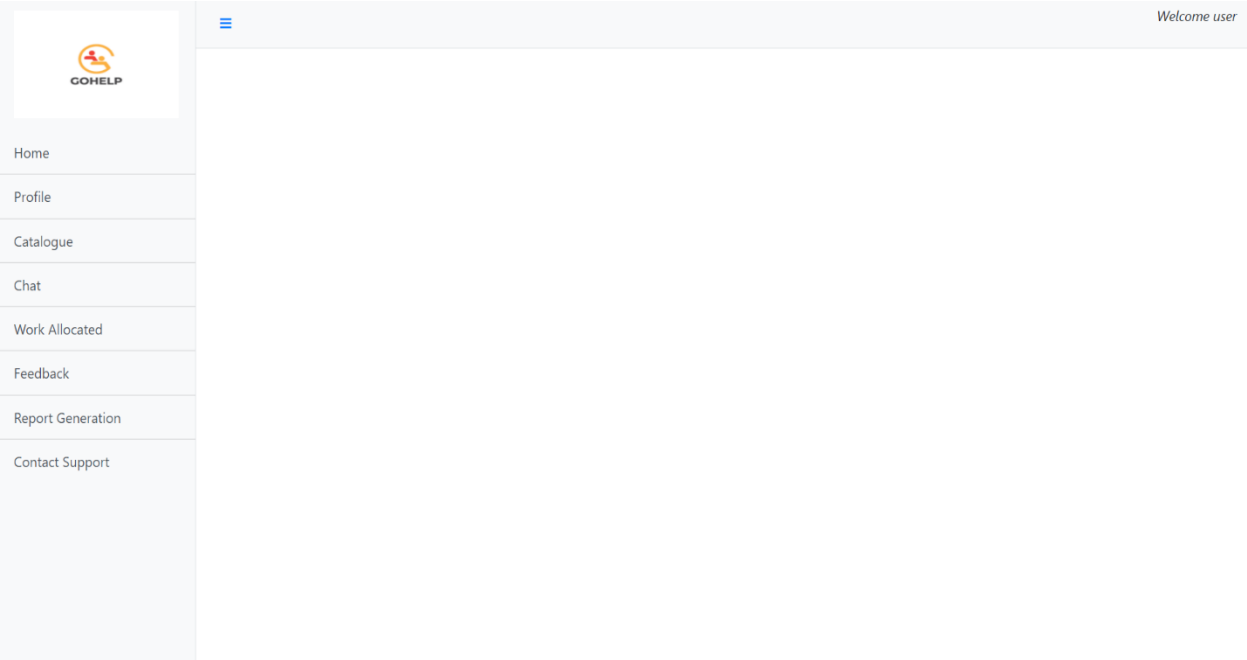
The overall architecture is displayed in the diagram as follows.



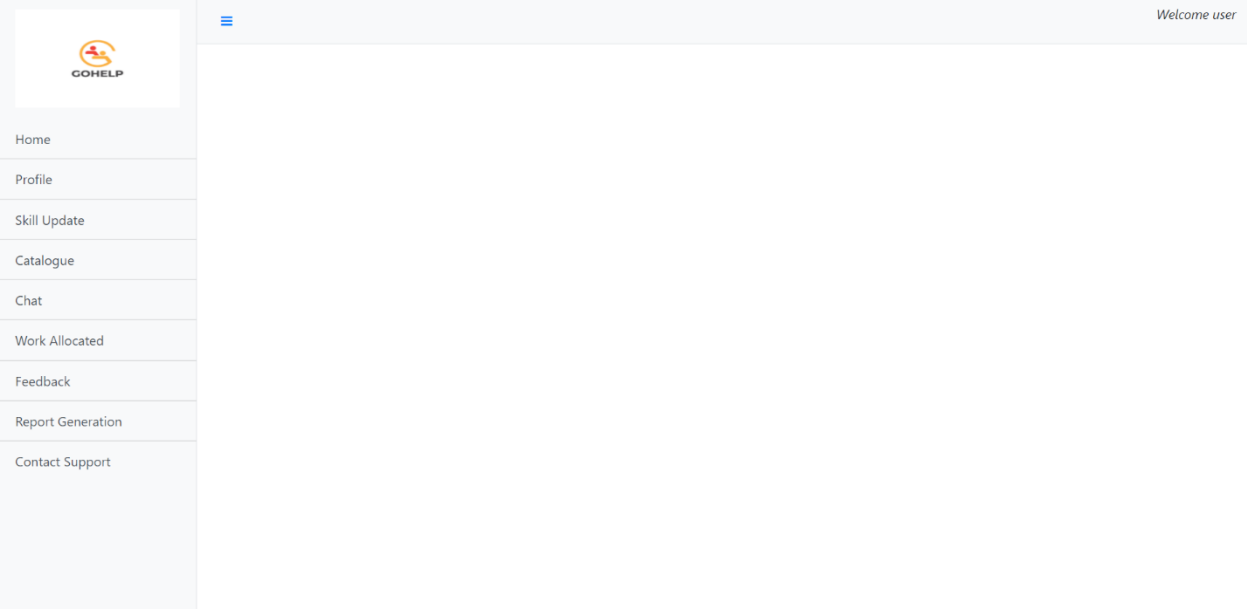
## 6.2 Screen Images

Some of the screenshots of the platform are given below.

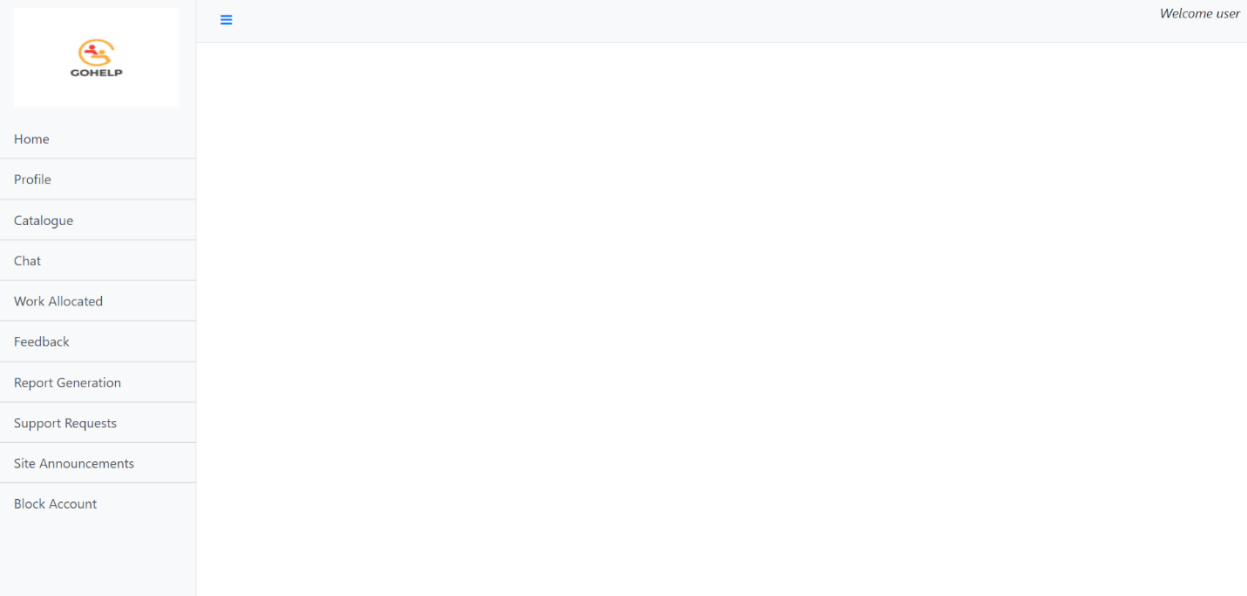
### Customer Dashboard



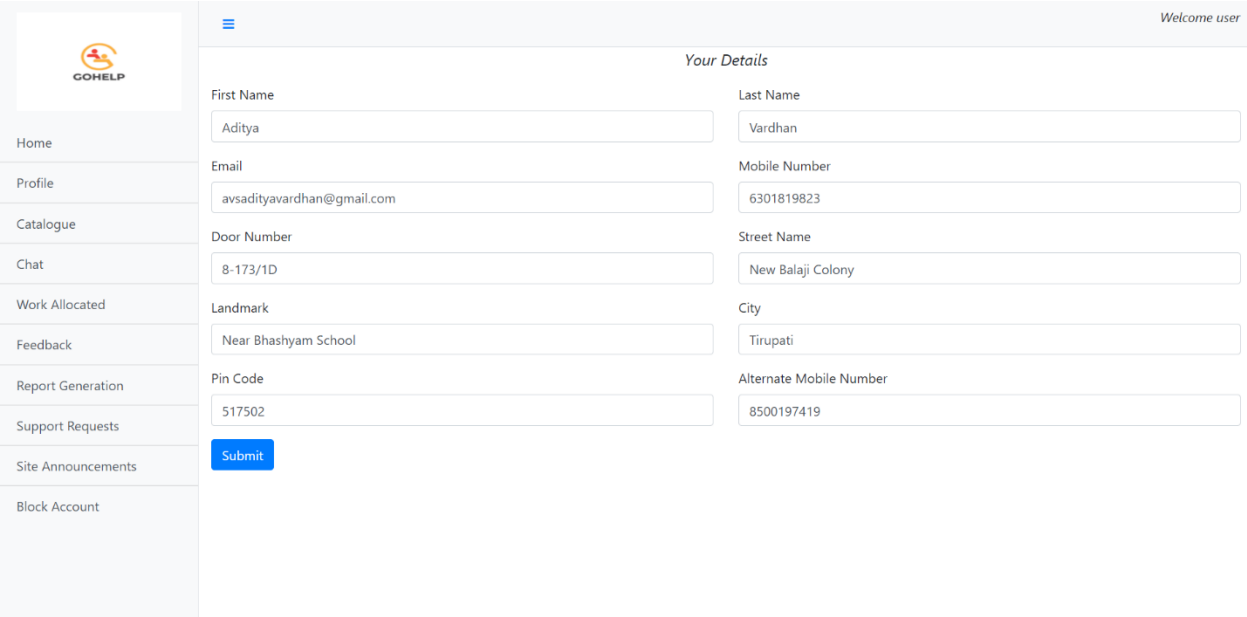

### Worker Dashboard



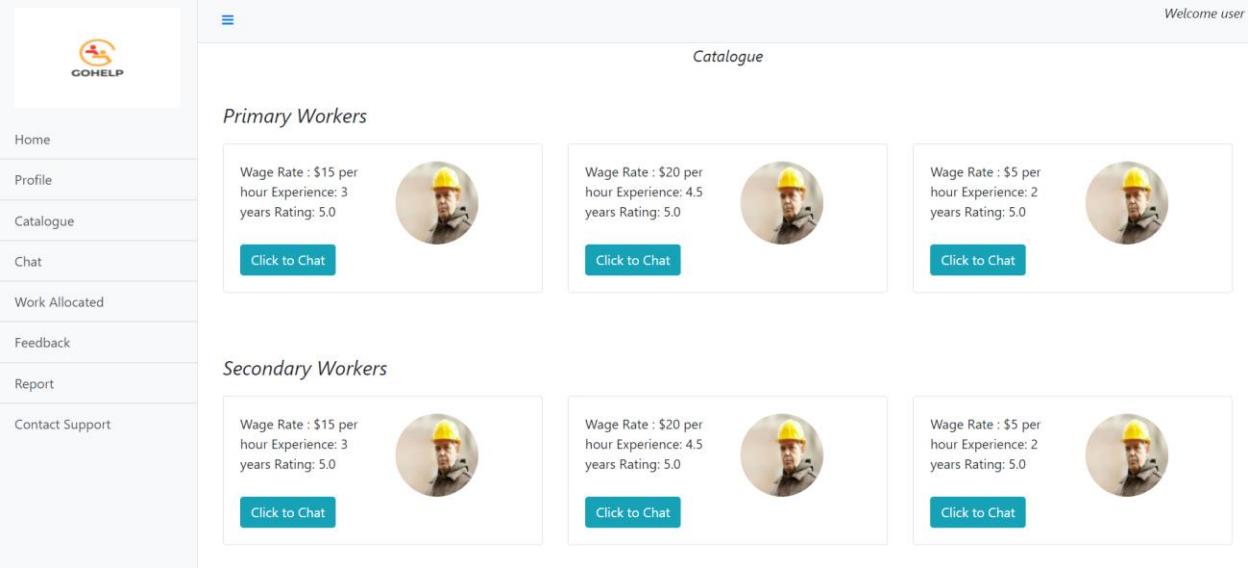
## Admin Dashboard



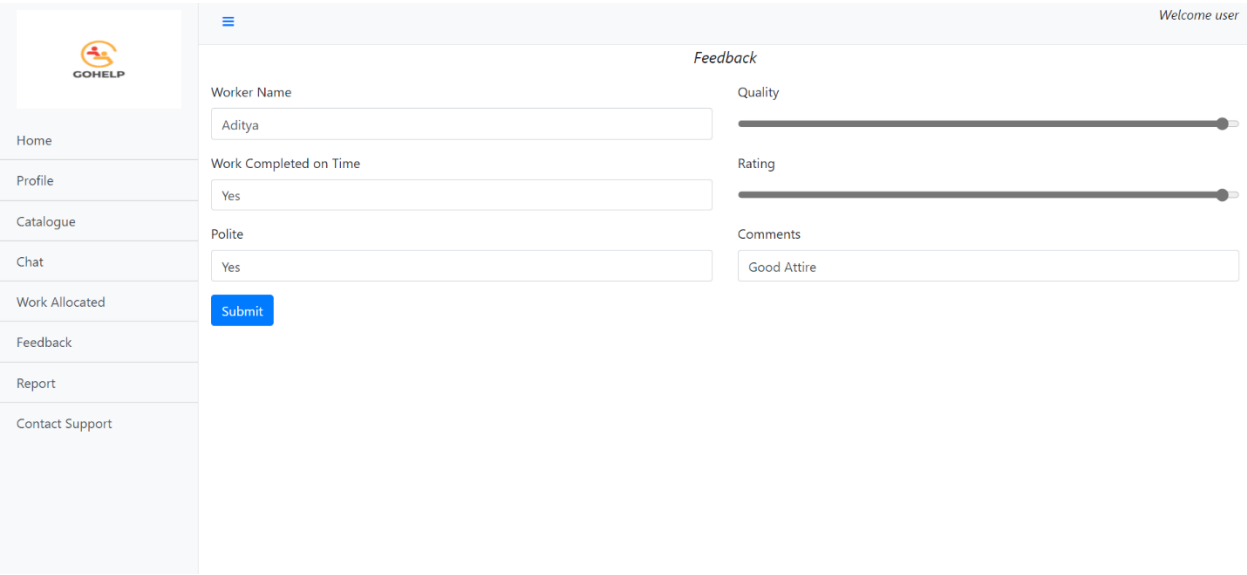

## Profile Form



## Catalogue:




## Feedback Form:



## 6.3 Screen Objects and Actions

Each of the objects associated with the screen performs the action mentioned below.

**Home Action:**

Home action shows a empty dashboard where user can select the next options mentioned in the dashboard panel. It is kept empty to add site announcements.

**Profile:**

On Clicking the Profile action, we will display the user details which he provided at the time of registration and can be updated by filling the data and clicking the submit button

**Catalogue:**

Customer on clicking the Catalogue button will be displayed with the primary skilled workers, secondary skilled workers where they can start communicating in the chat.

**Feedback:**

Customer can provide feedback to the worker for the work done by the worker. He can submit it on clicking the submit button.

All the other components will get the individual forms displayed in the dashboard.

## 7. Requirements Matrix

A matrix is constructed to keep track of the effect of changes happen when a record / table gets updated in the application.

| Functional Requirement | Description | System Components | Data Structure Involved |
|---|---|---|---|
| 4.1.3.1 Customer Registration | Registers Customer in the application | Registration | User, Profile |
| 4.1.3.2 Worker Registration | Registers Worker in the application | Registration | User, Profile |
| 4.2.3.1 OTP Validation | Validates the OTP received by user | Data verification and email notification | User |
| 4.2.3.2 OTP Resend | Resends the OTP | Data verification and email notification | User |
| 4.3.3.1 Login System | Performs user login | Authentication | User |
| 4.3.3.2 Forgot Password | Changes the user password | Authentication | User |
| 4.3.3.3 Remember Username | Helps to remember the username | Authentication | -(cookies in web browser) |
| 4.3.3.4 Logout | Logout the user | Authentication | User |

| | | | |
|---|---|---|---|
| 4.4.3.1 User Profile Update | Updates the User Profile | Profile and Skill Tab | Profile, User |
| 4.4.3.2 Skills Addition | Workers skill set is generated | Profile and Skill Tab | Skill, User |
| 4.4.3.3 Skills Updation | Workers skill set is updated | Profile and Skill Tab | Skill, User |
| 4.5.3.1 Display Catalogue | Display catalogue of skill set | Catalogue | Skill, User |
| 4.5.3.2 Wage Filter | Filters worker by wages | Catalogue | Skill, User |
| 4.5.3.3 Skills Filter | Filters worker by skills | Catalogue | Skill, User |
| 4.5.3.4 Ratings Filter | Filters worker by ratings | Catalogue | Skill, User |
| 4.6.3.1 Display Chat | Displays chat window | Chat | Chat, Message, User |
| 4.6.3.2 Close Chat | Closes the chat | Chat | Chat, User |
| 4.6.3.3. Fix Appointment | Setup the appointment | Chat | Appointment, Chat, User |
| 4.6.3.4 Change Appointment Timings | Changes the appointment timings | Chat | Appointment, User |
| 4.7.3.1 Customer Feedback | Customer feedback to worker | Feedback | Feedback_Worker, User |
| 4.7.3.2 Worker Feedback | Worker feedback to customer | Feedback | Feedback_Customer, User |
| 4.8.3.1 Block User | User blocks another user | Block and Report Generation | Block Request, User |
| 4.8.3.2 Admin Review | Admin Reviews the user request | Block and Report Generation | Block Request, User |
| 4.9.3.1 Customer Report | Report on customer | Block and Report Generation | Appointment, Feedback_Customer, User, Profile |
| 4.9.3.2 Worker Report | Report on worker | Block and Report Generation | Appointment, Feedback_Worker, User, Profile, Skill |
| 4.9.3.3 Admin Report | Report on both worker and customer | Block and Report Generation | User, Skill, Appointment, Feedback_Worker, Feedback_Customer, Chat, Block_Request |

# 8 Appendixes

No Appendix