Handyman App

Use Cases

Luke Boncich
Adam Campbell
Ajay Kc
Michael Tran

USERS

1. Signing into the App

Use-case Name:	Signing into the App
Actor:	User, Tasker
Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan application installed. 3. Verified HandyMan Account.
Flow of Control:	1. The user attempts to access his or her account with a login ID and password combination. 2. The system verifies that the login ID and password. combination is valid with a pre-existing account 3. The system checks whether the account is associated with a tasker or a user
Post-Conditions:	1. The user is given access to the app. 2. A connection to the server is established
Error-Conditions:	1. Login ID and Password combination is incorrect. Access to the app is denied. 2. The tasker/user account is suspended after 3 incorrect attempts. Access to the app is temporarily denied.
Non-Functional Requirements:	1. The application should be platform compatible 2. The connection with the server must be secure 3. The application must be in the preferred language chosen by the user (chosen upon first launch), or default English

2. Searching for a Tasker

Use-case Name:	Requesting Assistance
Actor:	User
Pre-Conditions:	1. Internet connection 2. Smartphone with in-built GPS feature and HandyMan Application installed 3. User is logged into the application. 4. User has a verified payment method
Flow of Control:	1. User selects the “Request Assistance” Menu Tab. 2. User selects the category of the task they need assistance in from the menu. 3. The user will fill out the task form, filling in the required information to property detail the task a. Task Title b. Description of Task c. Date of task d. Pictures e. Price/Rate 4. The User posts the request by pressing the submit. 5. The System displays a list of potential taskers,from which the user will choose the desired handyman.
Post-Conditions:	1. The handyman is notified by the system that he has been selected.
Errors:	1. No handyman is available within the GPS radius.
Non-Functional Requirements:	1. The application should be platform compatible 2. The connection with the server must be secure 3. The GPS feature should be turned on so that the server can display the nearby potential handyman.

3. Messaging Potential Taskers

Use-case Name:	Sending a Private Message to a Potential Tasker
Actor:	User
Pre-Conditions:	1. The user is already logged into the application 2. The user has searched for a tasker and can see a list of suggested taskers for the desired job.
Flow of Control:	1. The user selects the profile of the tasker that's to be messaged 2. On the profile, there exists a button called "Send Message." The user clicks this. 3. The user can now type the desired message to be sent to the tasker, and confirm the send. 4. The system filters the message to ensure it is safe to send, and sends the message to the tasker, or rejects the act of sending.
Post-Conditions:	1. The tasker received the message inside his or her inbox 2. The user is notified (by a small check symbol) that the message was properly sent.
Error-Conditions:	1. The user is notified of the reason why the message failed to send 2. Multiple offenses lead to a chat restriction
Non-Functional Requirements:	1. The connection with the server must be secure

4. Editing request

Use-case Name:	Editing a request
Actor:	User
Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan Application installed. 3. User is logged into the application. 4. User needs to have at least one request on his “Request Tab”.
Flow of Control:	1. User selects the “Request” button on the menu tab. 2. User selects the request that he wants to change. 3. The user will edit either of the following information: a. Task Title b. Description of Task c. Date of task d. Pictures e. Price/Rate 4. The user confirms the changes.
Post-Conditions:	1. The system updates the old request with new information and post the request. 2. Any taskers, who were interested in the request prior to the changes, are notified of the changes.
Errors:	1. The user leaves the required fields empty. 2. The date of task is a date before the current date. 3. The price is set equal to 0.
Non-Functional Requirements:	1. Some conditions should be set on user inputs so that it matches the criteria. 2. The date of the task and price is a required field.

5. Cancelling the request

Use-case Name:	Cancelling a request
Actor:	User
Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan Application installed. 3. User is logged into the application. 4. User needs to have at least one request on his “Request Tab”.
Flow of Control:	1. User selects the “Request” button on the menu tab. 2. User selects the request that he wants to cancel. 3. The user will cancel the request and provides the reason for cancelling the request.
Post-Conditions:	1. The system checks if any tasker has been assigned to the request or not. 2. The system cancels the request if a tasker has not been assigned to the request without any charges. 3. If a tasker is assigned, the system checks when the agreement was made and how late before the agreed date was the request cancelled. Finally,the system notifies the user of the charges for cancelling the request. 4. The charge amount is deducted from user’s bank account or paypal account. 5. The request is removed from the user’s list of request 6. The request is also removed from the database. 7. The user is notified of the cancellation.
Errors:	1. The connection to the server is lost.
Non-Functional Requirements:	1. The connection with the server must be secure.

6. Paying and rating the tasker

Use-case Name:	Paying the Tasker and rating the performance
Actor:	User
Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan Application installed. 3. User is logged into the application. 4. User needs to have at least one tasks on his “Current Tasks Tab” 5. User needs to be in agreement with some tasker with at least one of the current tasks. 6. Both user and tasker need to agree on the completion of task.
Flow of Control:	1. Both party, user and tasker, confirm the completion of task and sign on the timestamp. 2. The user decides to give some tip to the tasker. 3. The system charges the user for the service and displays it to the user following this procedure a. Initiation of Processing:The user initiates payment authorization request to their payment processor(Paypal,Visa, etc). b. Verification of Available Funds: The payment authorization request is successful and the payment processor sends a response to the system acknowledging that the funds are now held until the user finalizes the payment. c. Authorization of Transfer:the System sends a message back to the payment processor to finalize the payment. d. Completion of Transfer: The funds are immediately deducted from the users line of credit. The funds take 3 days to be transferred to the taskers bank account. 4. The user rates the tasker based on his performance and leaves a feedback.
Post-Conditions:	1. The system marks the request as completed and removes it from the database and user’s request list. 2. The system adds the service charge with the tip and sends the bank the final amount. 3. The system pays tasker 90% of the total amount paid by the user and keeps the 10% for the company.
Error-Conditions:	1. User payment method is declined(Insufficient funds) 2. Tasker payment is invalid(Expired information)

	3. Internet connection was lost while attempting to post review
Non-Functional Requirements:	1. The rating should be a 5-star system 2. Tip calculator with recommended percentages is integrated

TASKERS

1. Messaging a User as a Tasker

Use-case Name:	Sending a Private Message to a User
Actors:	Tasker
Pre-Conditions:	1. The tasker is already logged into the application 2. The tasker can see a list of suggested users requesting a task related to the tasker
Flow of Control:	1. The tasker selects the user's profile that is to be messaged 2. On the profile, there exists a button called "Send Message." The tasker clicks this. 3. The tasker can now type the desired message to be sent to the tasker, and confirm the send. 4. The system filters the message to ensure it is safe to send, and sends the message to the user, or rejects the act of sending.
Post-Conditions:	1. The user received the message inside his or her inbox 2. The tasker is notified (by a small check symbol) that the message was properly sent.
Error-Conditions:	1. The tasker is notified of the reason why the message failed to send 2. Multiple offenses may lead to either a chat restriction or a full qualification background check on the tasker
Non-Functional Requirements:	1. The connection with the server must be secure 2. Option to call recipient on the phone is available when viewing the message

2. Timestamping Jobs

Use-case Name:	(2a) Initiating the Beginning of the Task
Actors:	Tasker, User
Pre-Conditions:	1. The tasker and user have already agreed on the task to be performed 2. The tasker and user are present at the location the task is to be performed 3. The tasker and user are both logged into the application
Flow of Control:	1. The tasker selects the confirmed task on his or her list of tasks 2. After selecting the task, the tasker requests for the beginning of the task (by timestamp) and waits for user's confirmation. 3. The system pushes this request notification over to the user, who should be present with the tasker 4. The user confirms that he or she is okay with initializing the timestamp for the task
Post-Conditions:	1. The system reads that the user agreed on allowing the task to be initialized, and starts the timestamp accordingly.
Error-Conditions:	1. The system reads that the user did not agree on allowing the task to be initialized, and does not start the timestamp 2. The system sends a message to the tasker stating that the user did not agree with initializing the task, signaling miscommunication between user and tasker, or some other issue.
Non-Functional Requirements:	1. The connection to the server must be secure 2. The timestamp must be visible and clear to both the user and tasker

Use-case Name:	(2b) Ending the Timestamp for the Task
Actors:	Tasker, User
Pre-Conditions:	1. The tasker and user have already agreed on the task to be

	performed 2. The tasker and user are both logged into the application 3. The timestamp for the task needs to already be initialized and ongoing
Flow of Control:	1. The Tasker submits a request to place an endtime timestamp on the task 2. The system reads the request and pushes a notification over to the user stating the finalization of the task 3. The user may do nothing (accept that the task was finalized) or dispute the endtime timestamp if any issues occur 4. Under the condition that the user decides to dispute the finalization timestamp, the system notifies the tasker of this event
Post-Conditions:	1. The endtime timestamp signals the finishing of the task, so the system displays the total time it took to complete the task
Error-Conditions:	1. A user dispute was placed, marking the completion of the task as questionable
Non-Functional Requirements:	1. The connection to the server must be secure 2. The timestamp must be visible and clear to both the user and tasker

Use-case Name:	(2c) Timestamp GPS confirmation
Actors:	Tasker
Pre-Conditions:	1. Tasker and User agreed on job 2. Tasker has internet connection. 3. Tasker and User are logged in. 4. Tasker submits a time-stamp at start of job
Flow of Control:	1. System determines if Tasker is at GPS location of requested job 2. Timestamp initiation is completed (2a)
Post-Conditions:	1. User will know whether Tasker showed up at desired location. 2. Guarantees Tasker was at work-site based on GPS
Error-Conditions:	1. Could not connect to GPS thus can not find location 2. Tasker does not have internet connection
Non-Functional	1. Show history of time stamps if previous task has been

Requirements:	completed in same location in the past.

3. Searching for request placed by users

Use-case Name:	Searching for requests placed by users
Actor:	Tasker
Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan Application installed. 3. Tasker is logged into the application. 4. At least one request should be in the system.
Flow of Control:	1. Tasker selects the “Request List” from the menu tab. 2. Tasker narrows the search by entering his location, desired price, and date. 3. The system lists the applicable requests.
Post-Conditions:	1. After finding a desirable task, the tasker messages the user who made the request. 2. The system filters the message and notifies the user. 3. The user checks the message by the tasker. 4. The user messages the tasker to convey any additional information and possibly assign the tasker for the task. 5. The system notifies the tasker that he has been assigned of the task.
Errors:	1. The filters specified by taskers does not have any search hits.
Non-Functional Requirements:	1. The connection with the server must be secure 2. The GPS feature should be turned on so that the system can display the nearby requests. 3. There should be some validations set on the input by the tasker. For instance, he cannot search for past requests or completed requests. 4. Requests marked as completed should not appear on this list.

4. Receiving the payment and rating the user

Use-case Name:	Receiving the payment and rating the user
Actor:	Tasker

Pre-Conditions:	1. Internet connection. 2. Smartphone with in-built GPS feature and HandyMan Application installed. 3. Tasker is logged into the application. 4. Tasker needs to have at least one tasks on his “Current Tasks Tab” 5. Tasker needs to be in agreement with some user with at least one of the current tasks. 6. Both user and tasker need to agree on the completion of task.
Flow of Control:	1. Both party, user and tasker, confirm the completion of task and sign on the timestamp. 2. The system charges the user for the service and displays it to the user. 3. The tasker is notified of the charge on the user and the tip that the user gave. 4. The tasker rates the user and leaves a feedback.
Post-Conditions:	1. The system adds the service charge with the tip and sends the bank the final amount. 2. The system pays tasker 90% of the total amount paid by the user and keeps the 10% for the company.
Error-Conditions:	1. Tasker payment is invalid(Expired information) 2. Internet connection was lost while attempting to post review
Non-Functional Requirements:	1. Both, user and the tasker, need to have a valid bank account or a paypal account linked to their HandyMan app.

5. Viewing Profile as a Tasker

Use-case Name:	Viewing profile as a tasker
Actors:	Tasker
Pre-Condition:	1. Internet Connection 2. Logged into HandyMan application
Flow of Control:	1. Tasker presses the profile button 2. Bottom of profile menu will have reviews and ratings. a. Tasker can scroll through these but not edit them 3. Tasker can press ‘settings’ icon (a gear) to change personal information 4. Middle of profile will display recent work history.

Post-Condition:	1. If edits were made, profile will be updated 2. Update based on work-done / interactions with Users
Error-Conditions:	1. Not connected to internet 2. Failure to access credit card if entered incorrectly upon eidt 3. Photo resolution is too small when editing profile picture
Non-Functional Requirements:	1. Nothing will show for recent-history or reviews if Tasker has none 2. Tasker is officially registered as a Tasker

SYSTEM

1. Determining interface based on User / Tasker / Both

Use-case Name:	Determining interface based on User/Tasker/Both
Actor:	User, Tasker
Pre-Conditions:	1. Internet connection. 2. Smartphone with inbuilt GPS feature and HandyMan application installed. 3. Tasker and User are logged into the application
Flow of Control:	1. Following login, system checks database to see if the person logging in is registered as a Tasker or not. 2. The User (not-Tasker) logs in and will see a grayed out Tasker button. a. User presses grayed-out button and gets offered to register to become a Tasker i. App will be updated according if followed through 3. The Tasker will log in and the Tasker-menu button will be colorized and can be accessed normally.
Post-Conditions:	1. The Tasker-menu button will either work or not work according to runner of application. 2. Minor profile changes will happen: reviews / ratings displayed.
Error-Conditions:	1. Could not reach database to determine if User is also a Tasker 2. Internet connection drops during look-up. 3. User gets a working Tasker-menu button. 4. Tasker does not see Tasker-related profile information 5. Tasker is logged in only as a User.

Non-Functional Requirements:	1. Tasker needs to be registered as a Tasker before log-in.

2. Reviewing a User-Reported Issue

Use-case Name:	Reviewing a User-Reported Issue
Actors:	User
Pre-Conditions:	1. The user must have previously reported an issue in the form of a dispute 2. The user is logged into the application
Flow of Control:	1. The user selects from a drop-down list of options what the issue is, and submits this to the system 2. The system reviews the report, and flags the tasker for an offense related to the select issue option
Post-Conditions:	1. The tasker receives warnings on the first couple of valid disputes, with a possibility of removal from the app
Error-Conditions:	1. The user reporting sends too many reports, and the targeted tasker for the most part has a clean record. The user is flagged for disruptive activity and the tasker's warning is revoked
Non-Functional Requirements:	1. The connection to the server must be secure 2. List of issue options must contain the most likely issue options, and an "Other" option for custom typed issues not on the list