CMPE 277 - Project

University: San Jose State University (SJSU)

Course: Smartphone Application Development

Professor: Dr. Kaikai Liu

Team: ***15***

Name	Student ID
Shreya Ghotankar	015304393
Parvathi Pai	015293460
Sania Gonsalves	015313974
Krishna Jha	013803894

Table of Contents

Project Overview 3

App Features 3

Task Distribution 4

Implementation 4

GitHub Url: 8

Application Setup Steps - Android Studio: 8

Demo Link: 8

Code Readme: 8

Application Screenshots: 8

References 9

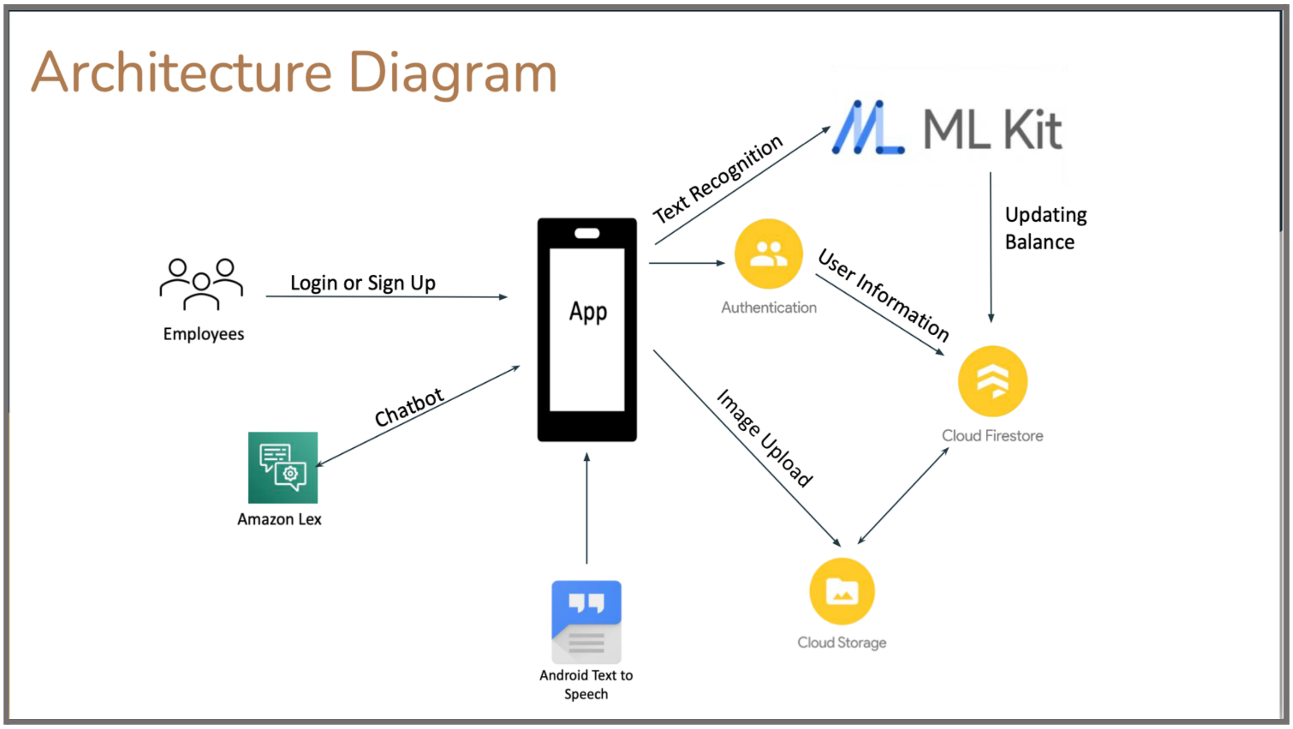
Project Overview

Due to the COVID-19 pandemic most of the companies have adapted work from home and would continue for an unforeseeable future. To help employees create a productive workplace at home and to have a good work-life balance, many companies are providing reimbursement on items like desks, food, Wi-Fi, gym equipment etc., that are purchased by the employees to sustain these unprecedented times.
In this process there are a lot of receipts to keep track off and it's very easy to misplace them. There are very few applications in the market which could help us to keep track of reimbursement.

iClaim provides a platform for easy enterprise reimbursements with smooth multi-cloud integration with AI technologies such as Text recognition, Speech processing and Chatbot.

App Features

- Firebase custom Authentication
- Upload image and save to Firebase Storage
- Text Recognition from uploaded image - using Google ML kit to identify the total amount.
- Firestore - using firestore to keep track of user balance and link user to the images stored in Firebase Storage
- Text to Speech - using android provided text to speech library
- Chatbot - conversational bot using AWS Lex



Task Distribution

Name	UI component	Backend Component
Shreya Ghotankar	App Icon, App bar, Dashboard, Add new claim page	Text Detection for the uploaded image using Google ML kit and update Firestore with claim information.
Parvathi Pai	User Authentication screen, Image upload, View Bills	Firebase custom Sign-in, Image upload capability and storing the image in Firestore/Storage in google cloud.
Krishna Jha	Chatbot screen	Using AWS Lex provides conversational interaction features to users and helps navigate the app.
Sania Gonsalves	Text to Speech button	Using android text to speech read claim information from the app screen and make the app convenient.

Implementation

Parvathi Pai

- Firebase Authentication
  - If the customer is using the app for the first time then he has to click on create account as shown here. This click leads the user to registration page as shown here. Link to PR
  - If the customer creates an account then his information is stored in Firebase Authentication as shown here
  - Then using the same credentials the user can login to the iClaim
- Polyglot persistence storage
  - If the user uploads the image (receipts), it will be stored in cloud firestore shown here
  - The metadata of the image and the user details such as username, user id will be stored in cloud firestore as shown here . Link to PR
- View Bills
  - Once the user uploads all the information from the previous steps.
  - The list of views will be displayed in the recycler view as shown here. Link to PR
- Logout
  - The user can logout by clicking on the logout button at the top right as shown here

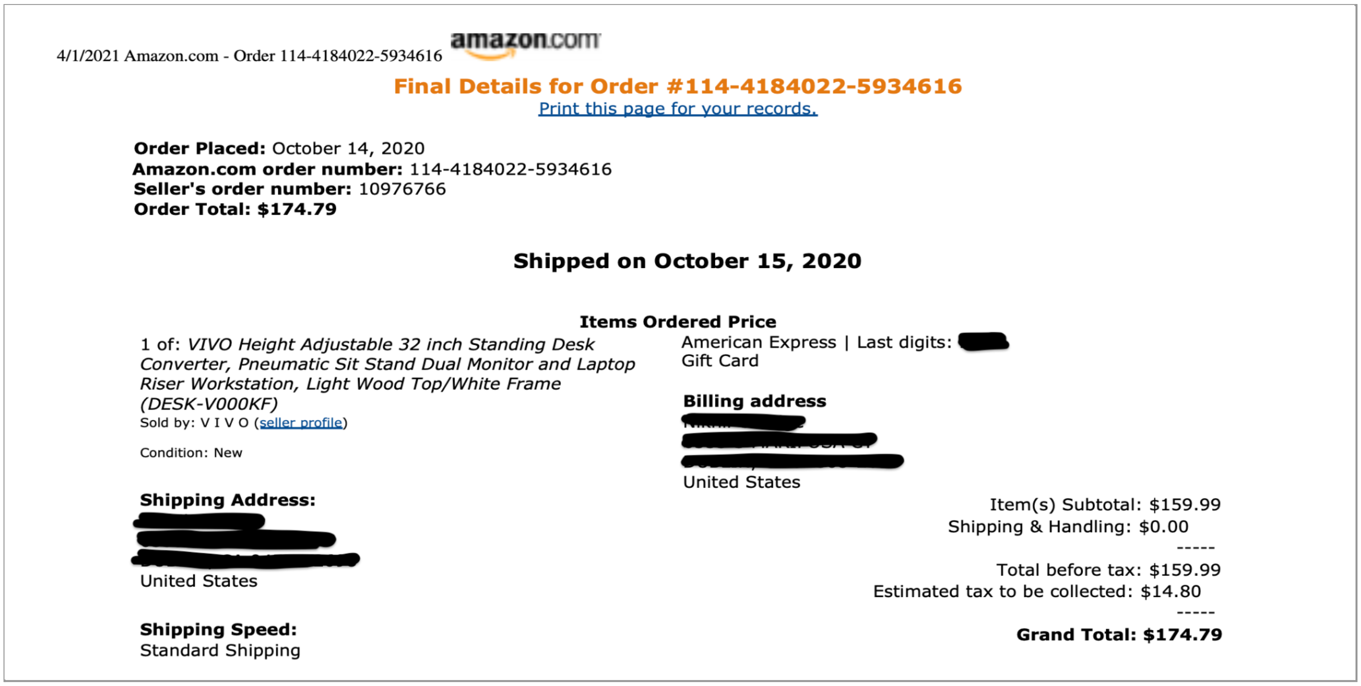
Shreya and Parvathi collaboration

- iClaim API
  - The iClaim API contains fields of the customer and the bill. This API can be extended to any number of entities.
- Add New Claim

- Created a form like page for users to enter bill related information like title, description and to upload image. Also there is a field which is populated with the total amount after text recognition is run on the image.

Shreya Ghotankar

- Created App icon using https://www.freelogodesign.org/
- Created App bar to add a signout button in Dashboard and Add new claim page. Also added up action button to all the activities layout.
- Dashboard
    - Added a watermark style logo as the background for the Dashboard activity.
    - Using the iClaim API, fetching the user name and balance to display on the dashboard page.
    - When a user is new the Annual balance is set to default of $500.00.
- Text Recognition
    - Using Google ML kit to detect the total amount from the uploaded bill image.
    - From the PostBillActivity.java onActivityResult method, the FirebaseVisionActivity.java is called after checking that the image is selected. In the FirebaseVisionActivity.java, getting the image uri and passing it to the TextRecognizer object using getClient().
    - For processing the text, reading each line using the class *Text.Line* and checking if the text contains **"Total"**. If a match is found, using that line to extract the bill amount for the claim and setting the detected line with **"Total"** keyword in the detectedTextView field.
    - Below is the example bill (redacted personal information), in this example bill image the "**Grand Total: $174.79**" line is detected by the Text recognizer.



4/1/2021 Amazon.com - Order 114-4184022-5934616

Final Details for Order #114-4184022-5934616

Print this page for your records.

Order Placed: October 14, 2020
Amazon.com order number: 114-4184022-5934616
Seller's order number: 10976766
Order Total: $174.79

Shipped on October 15, 2020

Items Ordered Price

1 of: *VIVO Height Adjustable 32 inch Standing Desk Converter, Pneumatic Sit Stand Dual Monitor and Laptop Riser Workstation, Light Wood Top/White Frame (DESK-V000KF)*
Sold by: V I V O (seller profile)

Condition: New

Shipping Address:
[redacted]
United States

Shipping Speed:
Standard Shipping

American Express | Last digits: [redacted]
Gift Card

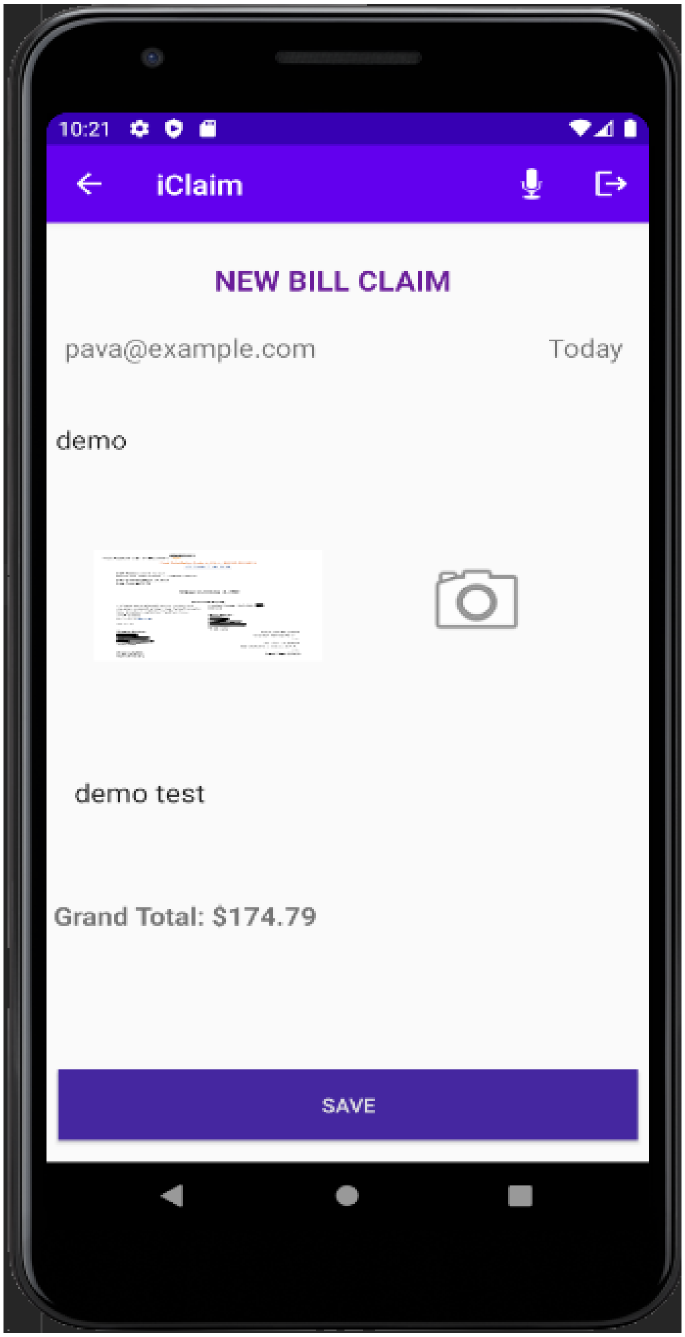
Billing address
[redacted]
United States

Item(s) Subtotal: $159.99
Shipping & Handling: $0.00

Total before tax: $159.99
Estimated tax to be collected: $14.80

Grand Total: $174.79

- In the screenshot below you can see that the text line “**Grand Total: $174.79**” has been set in the textview after successful text detection.



NEW BILL CLAIM

pava@example.com Today

demo

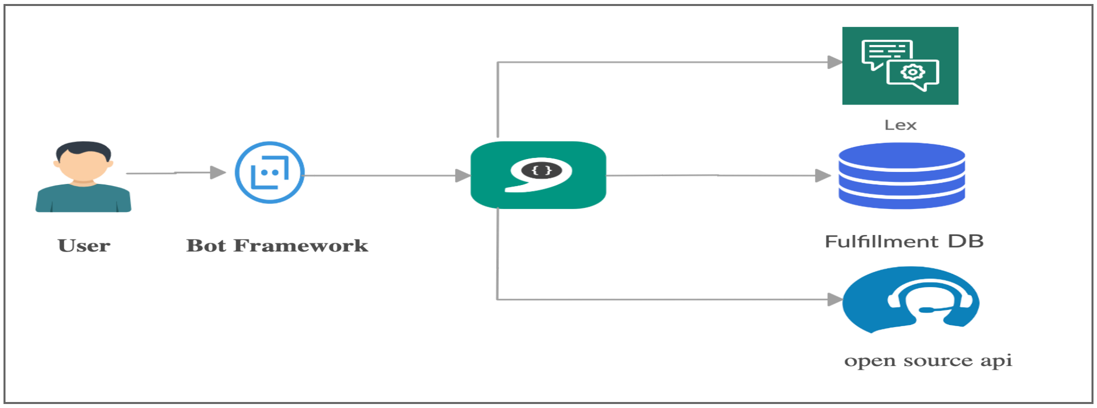
demo test

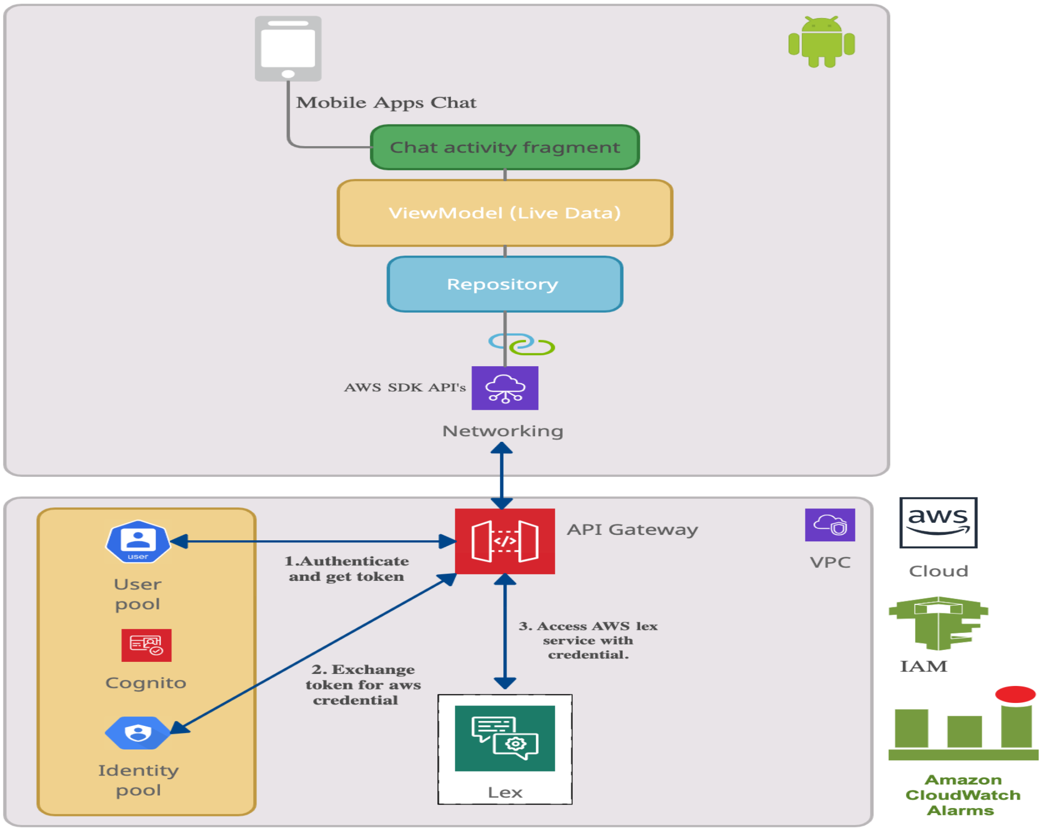
Grand Total: $174.79

- After detecting the required text from the image, calling [*adjustBalance* method](#) in [PostBillActivity.java](#). This method will extract the amount using regex pattern `("\\d+((\\.)(\\d{0,2})){0,1}")` and deduct it from the user's remaining balance amount and update it in Firestore so that when the user visits the Dashboard page again they see the correct value.

Krishna Jha

- Used AWS Lex for intent based conversation.
- Used AWS Cognito for OAuth2.0 feature. Please follow the steps for awsconfiguration.json file.
- BotActivity onCreate() is where the chatbot activity is initialized.
- init() internally called **initializeLexSDK**() to initialize the AWS Lex Client.
- AWSMobileClient is used for all auth related operations to accessing AWS Lex backend.
- startNewConversation() is where new the conversation initiation is happening.
- Multi-cloud strategy to bring unique capabilities of Google and AWS Cloud services.
- Future integration scope with fulfilment and third party tools like slack | Team etc.





Sania Gonsalves

- Text to Speech
  - Using android text to speech built-in API, implemented the text to speech which allows the application to read out the claim information from the App screen.
  - Add t2s to bill row - A class is added which is using inbuilt android TTS engine, that reads the iClaim username and bill information.
- Dashboard
  - Implemented a microphone speak button on the iClaim API at the top bar which will translate the text to speech using android built-in API and read out the Username and the Balance remaining displayed on the iClaim dashboard page.
  - View Bills Page - Added a microphone speak button on the View bills page to read out the description of the bill details uploaded by the user.
  - New Claim Page- The microphone speak button on the new claim page will read out the information about the username and the bill details.

GitHub Url:

https://github.com/ParvathiRPai/SmartPhoneApplication

Application Setup Steps - Android Studio:

https://github.com/ParvathiRPai/SmartPhoneApplication/tree/Documentation#how-to-setup-application-in-your-environment

Demo Link:

https://drive.google.com/file/d/1OWp3jGYVzEpzYkXNri37ipsZdaESnsac/view?usp=sharing

Code Readme:

https://github.com/ParvathiRPai/SmartPhoneApplication/blob/main/README.md

Application Screenshots:

https://github.com/ParvathiRPai/SmartPhoneApplication#ui-screenshots
https://github.com/ParvathiRPai/SmartPhoneApplication/tree/main/images

References

- https://firebase.google.com/docs/android/setup
- https://firebase.google.com/docs/auth/android/start?authuser=0
- https://developers.google.com/ml-kit/vision/text-recognition/android#java
- https://developer.android.com/reference/android/speech/tts/TextToSpeech
- https://aws.amazon.com/lex/
- https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-user-pools-authentication-flow.html
- https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-configuring-app-integration.html
- https://developer.android.com/samples
- https://docs.aws.amazon.com/sdk-for-android/
- https://docs.aws.amazon.com/mobile-sdk/index.html
- https://www.freelogodesign.org/