Rohan Aletty, Abhinav Ganesh & Arjun Anjaria
Team MANdroid
Unit 2 - Requirements

Application Description:

One of the most common issues that people face when attending company sponsored trips or interviews is keeping track of receipts that require reimbursement. Many times receipts are lost or people just lose track of when they should be getting reimbursed. Our app will solve that issue by allowing users to immediately document and submit their receipts in a quick and painless way.

Android Features:

During the design of our application we will be implementing the following seven features:

1) Camera
2) Camera – Autofocus
3) Camera flash
4) Accelerometer sensor
5) Touchscreen
6) Multi-touch screen
7) Scrolling touchscreen

We will be using the camera and all of its features such as autofocus and flash because we will need to take pictures of the receipts in order to document the transactions that have occurred. It is important that we have the focus feature because all of the information needs to be clearly displayed so that user's aren't reimbursed incorrectly. Flash is necessary in the case of having poor lighting during receipt submission.

Features 4 through 7 will be used as a primary means of design and navigation through the many screens of the application. The accelerometer sensor will be used for ensuring that the app can rotate between portrait and landscape modes. The various touchscreens will be implemented in such a way that the user will have to click on various widgets and buttons in order to perform the action of submitting a receipt. We will also include various scrolling features for use with the screens that require the user to look at multiple receipts.

Use Cases for GetMoneyGetPaid

1: Manage Receipts

Actor(s): User

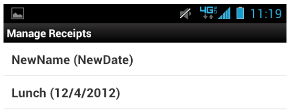
Description:
The user will be able to manage the receipts that he has yet to be reimbursed for. What we mean by management of the receipts is the actions of searching and viewing them. A user should be able to look up and see a specific receipt to see whether he or she has been reimbursed for it or not. Users should also be able to delete receipts that that are logged here (whether they have been reimbursed or not). Inside the details of the receipt, there should also be a button that allows the user to confirm that he/she has been reimbursed for the receipt.

Normal Flow of Execution:
The user hits the manage receipts button on the application and is able to see the receipts that have been added to the collection of his or her receipts. Next to the names of the receipts will be the date for which the expense occurred Furthermore, by clicking on the receipts, the user can get details of the transaction. In the details, the user either mark the receipt as paid, delete the receipt, or go back to the “Manage Receipts” screen.

Pre-Conditions:
(view) Home screen with “Manage Receipts” button.
(delete) Receipt in collection of receipts that has been reimbursed.
(mark paid) The receipt is marked as unconfirmed by the user.

Post-Conditions:
(view) Listing of receipts to be managed.
(delete) Collection of receipts should no longer include the deleted receipt.
(mark paid) In the future, the details screen will show the user confirmation of payment.

Wireframe of Screens:



Manage Receipts

NewName (NewDate)

Lunch (12/4/2012)

2: Payment History

Actor(s): User

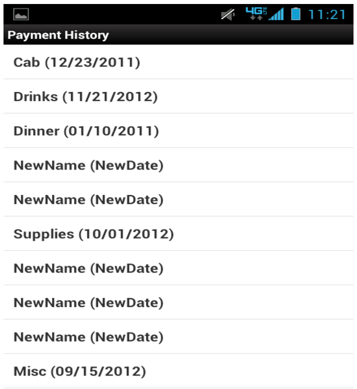
Description:
While it is the purpose of the application is to be able to manage the reimbursement of receipts, it would also be convenient to see the recent reimbursements that companies may have done for you. A listing of the recent transactions that companies may have completed could allow the user to view whether an expected reimbursement has been processed or not. This feature could also keep track of the amount of money still needing to be reimbursed. This feature would also allow you to delete reimbursements confirmations that you do not want to keep anymore.

Normal Flow of Execution:
The user hits the “Track Payments” button which navigates him to a screen showing the reimbursements that companies have done for him or her. He can view the details of the receipt that has been reimbursed for each payment that has occurred. The user can then decide whether or not he would like to delete a receipt that he sees on the list.

Pre-Conditions:
(view) The home screen with the “Track Payments” button.
(delete) The Payment History screen with the list of reimbursed receipts.

Post-Conditions:
(view) The screen with the most recent payments listed.
(delete) The Payment History screen with the list sans element deleted.

Wireframe of Screens:



Payment History

Cab (12/23/2011)
Drinks (11/21/2012)
Dinner (01/10/2011)
NewName (NewDate)
NewName (NewDate)
Supplies (10/01/2012)
NewName (NewDate)
NewName (NewDate)
NewName (NewDate)
Misc (09/15/2012)

3: Take Receipt Photo
Actor(s): User

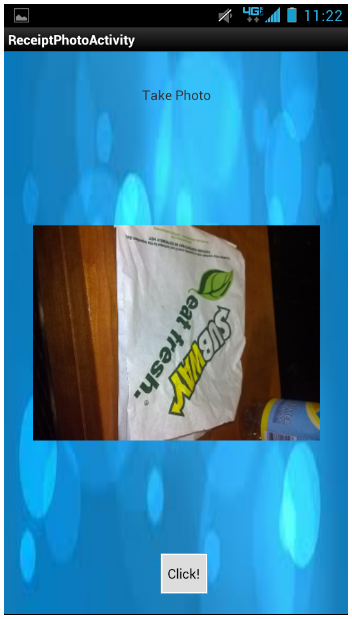
Description:
This allows the user to take a picture of a receipt for an expenditure they would like reimbursed. The application will interface with the built-in mobile cameras (front-facing and back) to let the user snap the picture. The user will be able to change flash and autofocus parameters to account for distance and lighting conditions. The user will have the option to delete and retake the photo if they are not satisfied with the quality of the photo. Finally, the user will have to ‘save’ the photo and provide a name/tag to reference it.

Normal Flow Of Execution:
The user will click the ‘Take Receipt Photo’ button on the home screen, to navigate to the ‘Take Receipt Photo’ screen. Here, the phone camera(s) will be activated, and the user can place the receipt appropriately to snap the photo. After the photo is snapped, the user will have an option to ‘Delete’ the photo or ‘Save and Continue’. If the user chooses to ‘Delete’, they will be navigated back to the ‘Take Receipt Photo’ screen. If the user chooses to ‘Save and Continue’, they will be prompted to provide a name/tag reference for the image. They will then be navigated to the ‘Manage Receipts’ screen.

Pre-Conditions:
(take-photo) Home screen with ‘Take Receipt Photo’ button.

Post-Conditions:
(take-photo) Camera-view, ready to take photo
(save) Photo is saved after snapped

Wireframe of Screens:



4. Submit Receipt

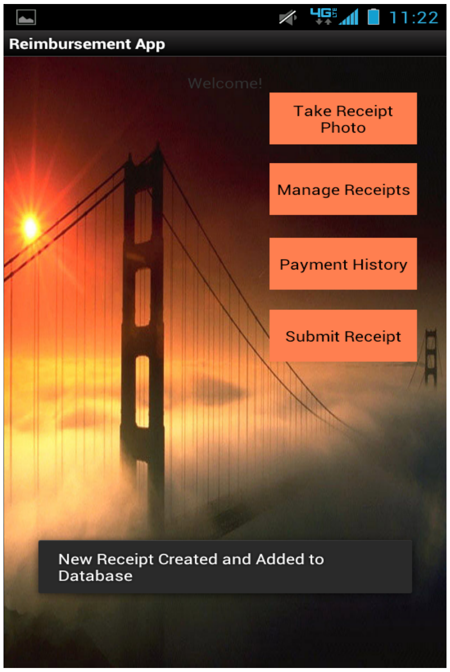
Description/Normal Flow Of Execution:

This button is in place to prove that we can eventually have the functionality of sending a receipt to a company. Currently when the button is clicked, a receipt is added to our database to show that our code can be extended to future iterations of our application.

Pre-Conditions:
(add receipt)	Receipt not in the Manage Receipts list.

Post Conditions:
(add receipt)	Receipt added to the database and will show up on Manage Receipts.

Wireframe of Screens:



New Receipt Created and Added to Database

Appendix I. Feature Considerations

The “Track Payments” and “Manage Receipts” use cases can be automated (with respect to tracking payments) by having a system whereby when the company has completed the reimbursement an automatic push notification or email is sent to the user notifying them of the completed transaction. Once this notification is sent to the user, his/her “Track Payments” screen would be updated with this transaction notice. The purpose of the “Track Payments” screen is to essentially act as a list of most recent completed reimbursements. The information gathered from “Track Payments” can be integrated with the “Manage Receipts” screen since “Track Payments” is a subset of the information in “Manage Receipts”. Another neat addition to the “Track Payments” screen would be to allow the user to sort payments by amount, priority, date etc. This would provide a convenient way for the user to process notifications.

Appendix II. Future Considerations (business side)

Our app basically provides a way to make reimbursements between companies and people more convenient. This means that we could approach the transactions from one of two angles. We could partner with the companies who are reimbursing to coordinate how they will pay people back. We could also partner with financial institutions that are associated with the various companies doing the reimbursing and set up a way to pay back through the banks themselves. If we partner with the companies, then they themselves can send notifications as to when they reimburse people. Furthermore, we can integrate the contacts of the people doing to reimbursing inside the company with the app so that any questions can be asked to them. Partnering with the banks could allow us to send notifications to the people being reimbursed immediately the money deposit goes through. This way, people can be sure that they have actually been reimbursed (rather than just a verbal confirmation through the company). There are also less

financial institutions to partner with than companies that could potentially reimburse clients. This is a big plus since many companies may use the same financial institution. After getting further with the implementation of our app, we will decide which set of groups would be the best to partner with. But for now, it is worth thinking about how the reimbursement confirmations will go through.