Requirements and Analysis for Good Deeds

Lo, Charles, Richard, Saga, Anton

2019-10-25

1 Introduction

The Good Deeds application aims to target the need for a service or a need to be a good Samaritan without any monetary transaction. The basic functionality is to be able to put out an ad-like post if you want to get help with something or if you want to help your fellow human beings. This should be shown to users in a way that is easy to navigate so that users can filter and find exactly what service they need.

The stakeholders in this application are anyone who needs help with something or wants to help others and want to do it in a more old fashioned way without any monetary transaction.

1.1 Definitions, acronyms, and abbreviations

User - A person using the application. The user can be either a client or a volunteer, depending on the users intentions. The user is not locked to be a client or a volunteer, and will change role if the intention changes.

Volunteer - A user that for this moment want to help others.

Client - A user that for this moment want to get help from others.

Receiving account - The client of the deed.

Giving account - The volunteer of the deed.

Deed - A listing made by users describing a service. As of right now, a Deed consists of a subject and a description. A deed can be either an Offer or a Request.

Offer - A Deed where a service is offered by a user. The user who created this offer will be registered as the volunteer by being the giving account. Other users can contact the volunteer and request to be provided the service.

Request - A Deed where a service is asked for by a user. The user who created this request will be registered as the client by being the receiving account. Other users can contact the client and offer to help with the service.

Claim deed - A user can claim a deed. If the deed is an offer, the user claiming the deed will be the client, by being registered as the receiving account. If the deed is a request, the user claiming the deed will be the volunteer, by being registered as the giving account. The deed will then be unavailable for others to claim.

Karma points - A user will use karma points as currency. When a person creates a new account, an initial amount will be assigned that new user.

Done deed - A deed is done when a claimed deed is marked as done. Karma points will then be withdrawn from the clients account and deposited into the volunteers account.

Marketplace - A page where users can see all the created deeds.

2 Requirements

2.1 User Stories

Epic

As a user, I would like to have an account, so I can manage my data.

Done:

Story Name: Create an account
As a user, I would like to create an account, so I can keep track of my deeds.

Tasks:
Create a "create account" button that leads to account creation
Create a view for account creation
Create a form in the account creation view
Create a submit button
Create a register button

Acceptance criteria:
The user is registered
Shouldn't be able to create the same account twice
Can't send the form without filling out the form
Validate fields

Story Name: Login
As a user, I would like to be able to log in to my account, so I can use my account.

Tasks:
Link "account-icon" to a login-page
Create a form for email
create a form for password
add login button that logs in the user
add cancel button that goes back

Acceptance criteria:
Only see the login screen before a successful login
Actually logged in

Story Name: Edit an account
As a user, I would like to be able to edit my account, since circumstances have changed.

Tasks:
Make the "login/account" button go to account-details view if logged in
Show all account details
Create an edit details view
In details, view get all fields that are editable prefilled
Add submit button
Add cancel button
Add back button

Acceptance criteria:
See correct data when editing account
Actually saves the updates
Validate fields

Story Name: Logout
As a user, I want to log out from my account, since I want to be able to log in to another account on my device.

Tasks:
Create a button
Create a method that logs an account out.
JavaDoc
Tests

Acceptance Criteria:
The account is logged out.
You're able to log in again.

Not Done:

Story Name: Delete account

As a user, I would like to delete my account, since I don't like the application anymore.

Tasks:
Add delete button
Add confirmation window
Add confirmation button on the confirmation window
Add cancel button on the confirmation window

Acceptance criteria:
The account is actually deleted
Only possible to delete my own account

Epic

As a volunteer, I would like to be able to offer my services, so that I can help others.

Done:

Story Name: Create an offer

As a volunteer, I would like to be able to offer a service, so I can help others.

Tasks:
Create button that leads to form
Create form
Create button that saves and closes the form
Create cancel button on form-view that closes the form

Acceptance criteria:
A user cannot submit a form without completing all the mandatory fields
Information from the form is stored
Cancel button does not save anything

Story Name: See my active offers

As a volunteer, I would like to be able to see all my active offers, so I can get an overview of my active offers.

Tasks:
Create a new view
Create a button to get to a new view
Create a list with all the users active offers
Create method to filter out offers
Have scroll on the view
Javadoc
Tests

Acceptance Criteria:
Only see my own offers

<u>Story Name: View an Offer</u>
As a volunteer I would like to view my offer, since I want to check that it looks okay.

Tasks:
Create a "View Offer" button/clickable that opens the offer.
Create a view for viewing offer.

Acceptance criteria:
The offer shown matches the offer pushed.

<u>Story Name: Delete an Offer</u>
As a volunteer, I would like to be able to delete my offer, since I can no longer offer that service.

Tasks:
Create a delete button
When button is pressed the data is deleted

Acceptance criteria:
Cannot deleted already done deeds
Can only delete you own offers
Deleted offers should disappear

<u>Story Name: Edit an Offer</u>
As a volunteer, I would like to be able to edit my offer, since the circumstances have changed.

Tasks:
Create a button that leads to edit form
Create an edit form
Create a button that updates the data
Create a cancel button that closes the form

Acceptance criteria:
Offer is updated
Cancel button don’t change anything
Cancel button closes the form
Can only edit your own offers

<u>Story Name: No access</u>
As a volunteer, I want information on why I can't create an offer when I'm not logged in, since I want to be able to create an offer.

Tasks:
Add a view for info on why offer can't be created
Make "new offer"-button lead to that view
Add a button on that view tog get to login page

Acceptance criteria:
"new offer"-button should lead to "no access" page only if user is not logged in

Epic

As a client, I would like to be able to ask for help, so that I can get help from others.

Done:

<u>Story Name: Create Request</u>
As a user, I would like to be able to create a request, so that I can receive services from others.

Tasks:
Create a request button
Refactor to use Create Offer view

Acceptance criteria:
Create Offer should work as before, with added functionality to choose a request

<u>Story Name: See all my requests</u>
As a client, I would like to see all my active requests, since I may want to interact with them.

Tasks:
Add methods that gathers information about the requests and passes it to the view
Create button to get to view with all requests
Create view for showing all requests of the current user
Show all the active requests
Add a way to press an active request that enables you to get to that request
Add "go back" arrow to get back to the view before this one

Acceptance criteria:
Should only be able to see own requests

Story Name: See my request

As a client, I would like to view my request, since I want to decide if I want to interact with it.

Tasks:

Create view for a request
Add a path to get to that request

Acceptance criteria:

Only the pressed request should be shown

Story Name: Edit my request

As a volunteer, I would like to be able to edit my request, since the circumstances have changed.

Tasks:

Create button that leads to edit form
Create a edit form
Create an update button that updates the data
Create a cancel button that closes the form

Acceptance criteria:

Request is updated
Cancel button doesn't change anything
Cancel button closes the form
Can only edit you own requests

Epic

As a user, I would like to be able to see all active deeds, so I can see who offers or ask for help.

Done:

Story Name: Marketplace

As a user I would like to have a marketplace, since I would like to see all available deeds.

Tasks:

Create a market view
Make it possible to see all offers
Make it possible to see all requests
Create a button which leads to the view
Add a text that says Browse Deeds
Create a button for showing all offers
Create a button for showing all requests

Create javadoc for new methods
Create tests for all new methods

Acceptance criteria:
Can see all requests when logged out
Can see all offers when logged out

Epic

As a user, I would like to change the status of a deed, so everyone knows if it is or will be done.

Done:

<u>Story Name: Claim a Deed</u>
As a user, I would like to be able to claim a deed, so no one else can choose the same.

Tasks:
Add a claim button to view deed
Add logic in model for checking if a deed is claimed
Add logic in model for checking that the logged in account is not the creator of the deed
Add logid for claiming the deed

Acceptance criteria:
Button only visible if deed is not already claimed
Button only visible if the logged in account is not the creator of the deed
Once pressed, deed becomes claimed
Cannot claim your own deed
Deed is not shown on marketplace once it's claimed

Not Done:

<u>Story Name: Done Deed</u>
As a user, I would like to be able to mark a deed as done, so It's clear that it's not active.

Tasks:
Add a done button

Acceptance criteria:
Button only visible if deed is claimed, but not done
Done deeds don't show on marketplace
Done deeds don't show in my claimed deeds

Cannot claim a done deed
Cannot delete a done deed
Cannot edit a done deed

Epic

As a user I would like to get karma points for doing good deeds, so I can show off with my kindliness.

Not Done:

Story Name: Add karma points to offer

As a user I would like to be able to add how many karma points I want for helping someone else.

Tasks:

Add karma points as a variable to deed
Add field for entering amount of karma points when creating an offer
Add variable for total amount of karma points to account
Add logic for getting the points if you are a givingAccount
Show total karma in accounts view
Add claimed deeds to accounts view

Acceptance criteria:

When done is pressed the account receives points for the deed

Epic

As a user I would like to give karma points to those who help me, so I can show my gratitude.

Not Done:

Story Name: Add karma points to request

As a user I would like to be able to add how many karma points a request will give.

Tasks:

Add field for entering amount of karma points when creating a request
Add logic for decreasing receivingAccounts amount of points when request is done

Acceptance criteria:

When done is pressed the users total karma points decreases

Epic

Technical Debt**:** Refactoring, optimization & small improvements.

Done:

<u>Story Name: TOAST</u>

As a user I would like to get a response, when clicking on a button, regardless.

Tasks:

Create a toast, that shows if the view is empty.

Acceptance Criteria

Be able to delete deeds and then get a toast
Your own deed view shows a toast if it's empty
Marketplace gives a toast if it's empty.

<u>Story Name: IAccount</u>

As a Developer I want an interface for account, to follow design patterns.

Tasks:

Create an interface
Implement the interface

Acceptance Criteria:

It works as it did before.

<u>Story Name: Javadoc Public Methods</u>

As a dev, I want javadoc so I know what the public methods does.

Tasks:

Add Javadoc to all public methods

Acceptance criteria:

All public methods should have javadoc

Not Done:

<u>Story Name: Remove current deed</u>

As a dev, I don’t want to store the currentDeed in the model, since its for the views.

Tasks:

Remove variable from goodDeeds
Remove unnecessary methods
Make views work anyway

Acceptance criteria:

Everything still works
All tests go through

Story Name: Data-management
As a dev, I want to be able to manage data, since I don't want to lose data between sessions.

Tasks:
Add logic for storing new data
Add logic for getting stored data
Add logic for deleting stored data
Add logic for modifying stored data

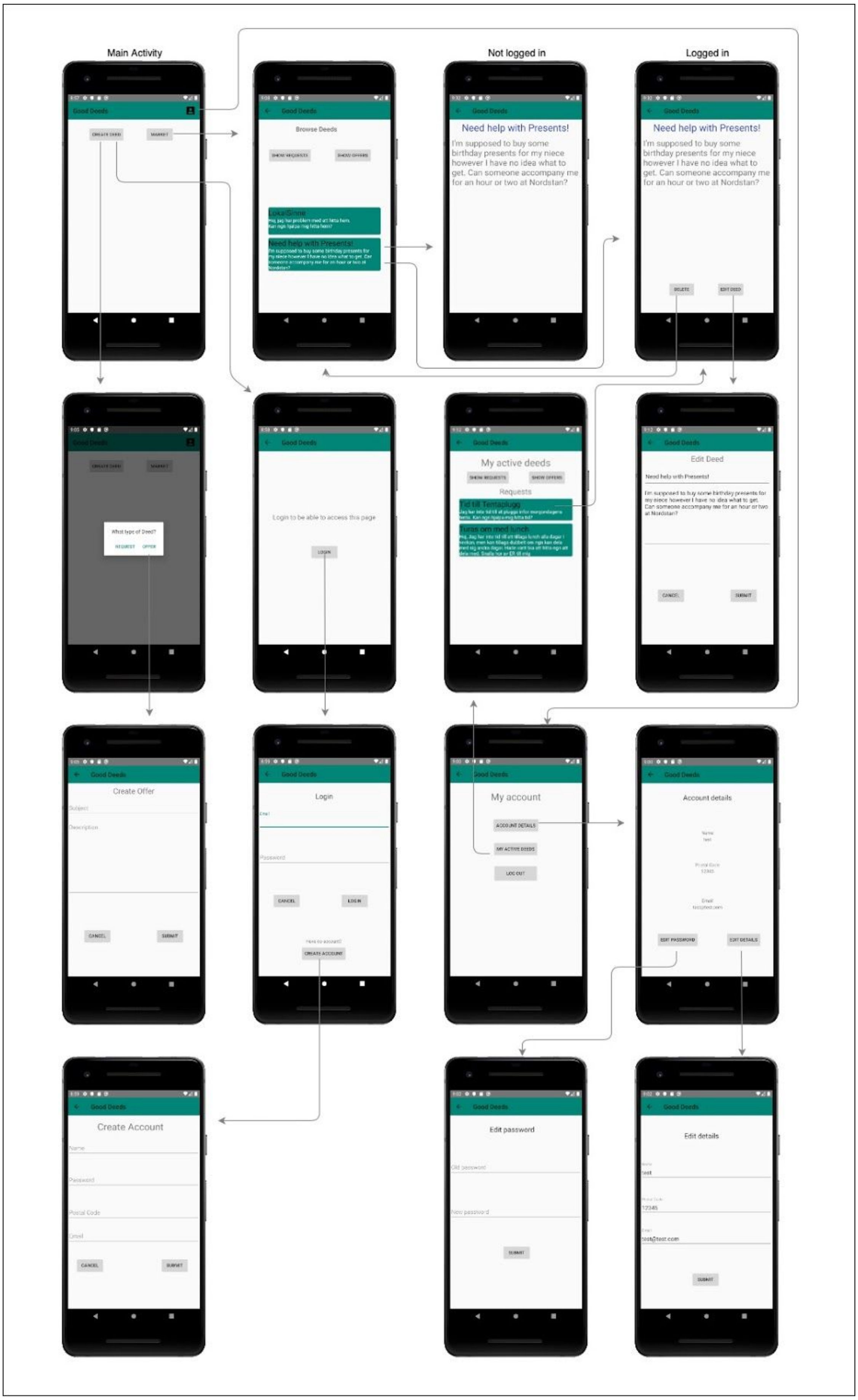
Acceptance criteria:
New data should be stored correctly
The data should be correctly read from the storage
Edited data should be stored correctly
Removed data should not be present in stored data

2.2 Definition of Done

- Unit tests should all pass
- Code should have been reviewed by at least two developers
- Acceptance criteria should be met
- Code should be merged with master
- The user story should be marked as complete
- Domain design should be reviewed and updated if any change

2.3 User interface

Buttons “login” and “submit” always leads back to Main Activity, while “cancel” leads back to the preceding view from where the button was pressed.

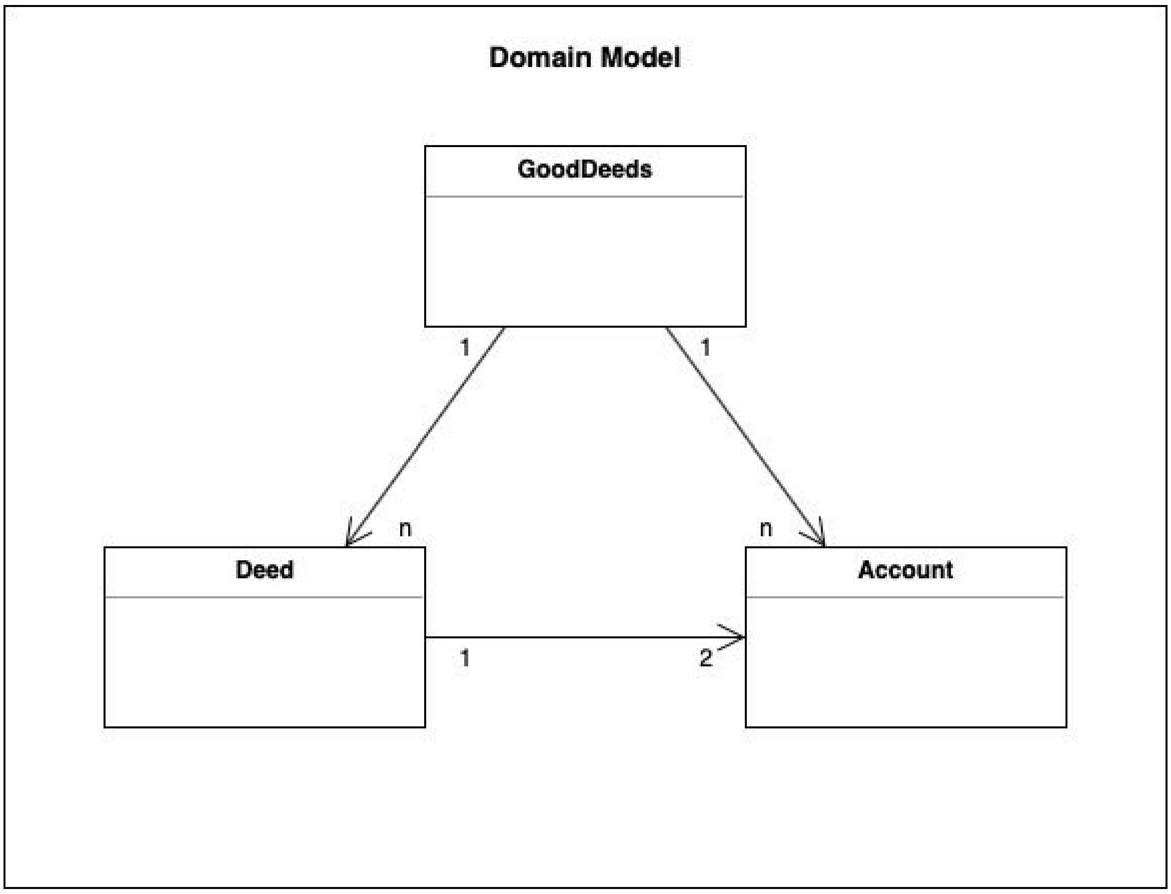


Hej, jag har problem med att hitta hem. Kan ngn hjälpa mig hitta hem?

I'm supposed to buy some birthday presents for my niece however I have no idea what to get. Can someone accompany me for an hour or two at Nordstan?

Name: test
Postal Code: 12345
Email: test@test.com

3. Domain model



3.1 Class responsibilities

-Deed
Represents a deed, which can be either a request or an offer. Keeps track of who is volunteering or getting help by variables givingAccount or ReceivingAccount. A deed can be either an offer or a request. Type of deed is set by initializing either givingAccount or receivingAccount to the logged in account as the deed is created.

-Good Deeds
Acts as the model for the project, holds data which views might need to have or methods that's necessary for the project to work properly. Works as an aggregate and holds a list of existing deeds and accounts.

-Account
Represents an account. Initializes the fields: name, postalCode, email and password when an account is created.

4 References

- Json - data interchange format http://json.org/
- RESTful api - architectural style for distributed hypermedia systems https://restfulapi.net/
- Travis - a continuous integration tool used to test software projects at github https://travis-ci.org/
- Github - a hosting service for software development version control using git https://github.com/