## Part 0

We met with TA on Nov. 4. The TA suggested that we should totally change the idea which is not commercial. So, we find a new idea about books which is user central. And it contains two types of users (Host of channel and Participants) who have different types of view of the system and interact differently. Also, we try to use APIs in real time by searching books from Open Library. Finally, we follow the suggestion and add Google and Facebook link for logging in.

## Part 1

* 1. *What will you project do?*

Our project is to implement a web application that helps people know the books they want to read and the link to buy it. Besides it gives people a chance to rate for the books they have already read. Also, it will give people a chatting channel to discuss about the books they read.

* 1. *Who or what will be its users?*

Our user is people who want to find books which they are interested in or who want to rate some books they have already read. And our user can also be a host of channel who has the authority to manage his channel. Or our user just wants to participate in a channel and discuss about the book.

* 1. *What do you think you'll be able to show in your demo?*

We will be able to show the structure of our website. The user can sign up and log in the web. We will also try to implement some interaction design of our service.

* 1. *What kind of data do you plan to store?*

We plan to store the book information the user searched as the searching history and our book database and store the rate information the user gives. And we will store the user’s data, such as username, password, channels they hold or participate in. Then we will also need to store the information they discuss.

* 1. *What API do you plan to use and what will you use it for?*

We plan to use APIs, such as Books, Google Map API, Amazon Product Advertising API, Calendar, Communication (MailGun), Google and Facebook API for Login and maybe some Machine Learning APIs.

## Part 2

*2.1 As a user, I want to search a book in the website so that I can find the information of the book, like the category of the book, the URL to buy it online and the rate from other users.*

My condition of satisfaction is that I give the right book name and click the search button. And there is book related to the name in the API or our database.

*2.2 As a user, I want to create a channel in the website so that I can have a personal place to invite people who also read the book to discuss about the book.*

My condition of satisfaction is that I click the ‘Create a Channel’ button and give the name of the channel.

*2.3 As a host of channel, I can invite people or accept joining request to the channel I own so that I can discuss about the book with my friends.*

My condition of satisfaction is that I click the invite or agree button to give others the permission to join in the discussion channel.

*2.4 As a normal user, I can send the joining request to the host of the channel so that the host of channel can know that I want to join his or her channel.*

My condition of satisfaction is that I have already chosen the channel which I want to join in. Then I input some request text and click the ‘send request’ button.

*2.5 As a user, I can rate for the book which I have already read for more than one day so that I can express my opinion for the book.*

My condition of satisfaction is that I have read book and signed in the calendar for more than one day. And then I click the ‘want to rate’ button and input all the rate information, like rate for plot.

*2.6 As a user, I want to get the recommendation about the books which I am maybe interested in so that I can read new interesting books.*

My condition of satisfaction is that I have logged in the website and have already choose the category I am interested in. Or I have searched some books in the website with an account.

## Part 3

*3.1 A user logs in the website, inputs the book name in the search bar and clicks the search button.*

Common case: The input is a http request with body. The output is a response with the description and ratings about the target book.

Special case: The book name is not in our database or the user does not input a book name. The output is a message to remind the user to input a valid book name.

*3.2 A user clicks the “creating a channel” button and inputs a channel name.*

Common case: When the user clicks the “creating a channel” button, the client sends a request. Then server responses with a dialog box to allow the user to give a book name. And when the user inputs a channel name, the client sends a request with a body including the book name and the server redirects the user to a chat room about the book.

Special case: The book name is not in our database. The output is a message to remind the user to input a valid book name.

*3.3 A host user clicks the “accept” button to allow other users to join in the chat room or clicks the “invite” button to invite other users.*

Common case: When the host clicks the “accept” button, the client sends a requests with a body including the user name of the applicant and the server redirects the applicant to the chat room. When the host click the “invite” button, the server responses with a dialog box for the host to give a user name. After the host inputs a user name, the server sends a invitation response to the invited person.

Special case: When the host clicks the “invite” button, the host inputs an invalid user name. We return a message to remind the host to give the valid user name.

*3.4 A user clicks the “join” button of a channel and inputs some texts to the host.*

Common case: When the user clicks the “join” button, the client sends a request with body including the user’s name, the host’s name, the texts and so on. Then the server return a response with the applicant’s information and request texts to the host.

Special case: The channel the user wants to join dose not exist. We return a response to remind the user that the channel has not been created yet.

*3.5 A user signs in the calendar more than once, and clicks the “rating” button and rates the book.*

Common case: When the user signs in the calendar, the inputs are some http requests with body of signing information. We stored the body information in the table of users. When the server receives a request to rate, it will check whether the user is qualified to rate the book. If the user is qualified and rates a book, we will store the user’s rating points of the book.

Special case: The user clicks the “rating” button before he or she already signs more than once. We will return a message to remind the user to sign in the calendar.

*3.6 An old user has searched several books or chosen his or her favorite categories.*

Common case: The input are some book names and genres the user is interested in from our database. The output is a response with some recommended books based on the algorithms of APIs.

Special case: The book name is not in our database and the user does not choose a category. The output is a response with several best sellers.

## Part 4

We plan to use JavaSE-11 as the compiler, Eclipse as the IDE, Maven as the build tool, CheckStyle as the style checker, JUnit as the unit test tool, Emma as the coverage tracking tool, Spotbugs as the bug finder and SQLite as the persistent data store.

## General information of Team

Team name: iDrop

Team member: Feiqiang Shen (fs2693), Hankun Cao (hc3153), Tianyi Wang (tw2677), Miao Liu (ml4410)