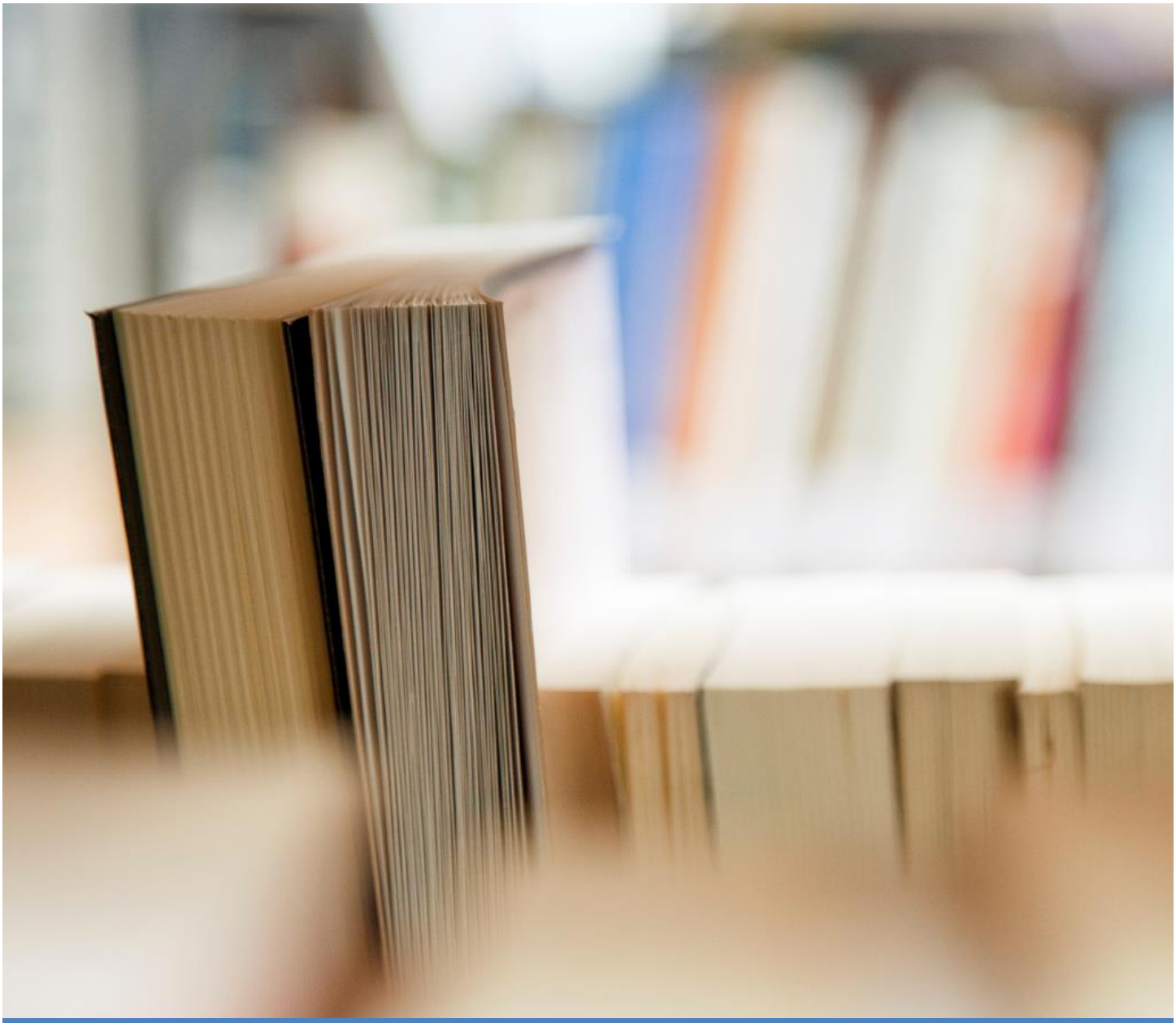


CS2003 Usability Engineering – Book Finder

Group 27 – Tzu-Lun Yeh 1814172, Franco Evett 1814746, Pruthvi Lalji 1813459, Anas Zouhir 1835227, Felipe Cardoso 1841029, Camron Darpoh 1839746, Romi Patel 1813708

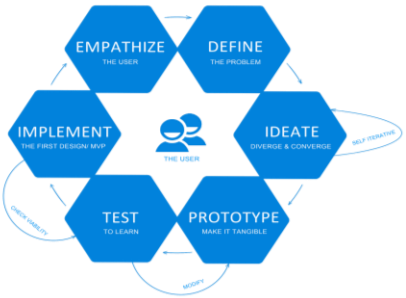
Video Link -
https://youtu.be/AAgVm0phL-c

Identifying User Requirements

Prior to making any prototype, we had to find out what are user requirements. We drafted a quantitative survey for our fellow CS2003 students. We choose quantitative based survey, as we wanted to calculate average of the user's input and identity, what area needed to be improved upon. Also, as group we brainstormed and identified all the requirements that we could think of.

Star Methodology

We used star methodology as our project's evaluation centric, where we plan to put our prototype in cycle and keep repeating until we get a finalised prototype with minimal or no errors. Following are some steps that we carried out under star methodology; Group meeting, prototype surveying, evaluating feedback and improving prototypes based on feedback.



Recruiting Evaluators:

Selecting our evaluators was simple as we were only allowed to collect data from our fellow course members, but we still tried to get an even sample of ages and genders to create as unbiased a collection of data as possible. Our course members are also familiar with the evaluation process. Our evaluators are familiar with the type of system our app is built for, we have selected a group of individuals that have experience using libraries and library apps to enable a smooth evaluation process.

Methods of Evaluations:

Questionnaire

We drafted a questionnaire with 10 questions asking users to review library experience. We asked both open ended and closed ended question to get quantitative and qualitative data. We believe getting both types of data will help us understand user's needs quickly, as well as get better understanding of what features needs to be worked on.

Observation

Task Checklist		
Sr. No	Instructions (Physical Task)	Tick ✓
1	Sign into the app	
2	View main menu books	
3	View recommended books	
4	Select a Book	
5	Look at the review and/or review the book	
6	Locate the book	
7	Search for a book	
8	Log out	

We used observation as one of our evaluation methods to evaluate our prototype. Evaluators were instructed complete set of tasks and tick them as they complete it. Evaluators were also asked to think out load while completing tasks. We observed while evaluators were completing the tasks. At the end of each task we asked four questions (Refer to end of Task question table) and noted down their answers. We got very good feedback from users straightway about difficulties were facing.

End of Task question	
1	What are you doing?
2	Why are you going this?
3	What are you expecting to happen?
4	Do you like this feature?

Heuristic Evaluation:

Establishing Heuristics:

The first step was to develop a set of testing category's (heuristics) for us to compare iterations of our app in and be able to collect user opinions and data. The full list of our categories is in the heuristic table below this section. Some of our chosen heuristics are: **Aesthetic and Minimalist Design, Flexibility and Efficiency of use** and **Help and Documentation.** These choices were made from our market research and our own observation of other similar apps such as amazon, Netflix, and other book apps.

Briefing Evaluators:

We have created a standardised brief for our evaluators to make sure they are all told the same things to do and to test. This will ensure none of our results are biased. We will in this case give them a set of features based on our heuristics list to evaluate and provide feedback on. This will create a narrower focus for us to work on than if we let them have free rein to test all functions of the app. This will enable us to focus on certain aspects of each prototype for each iteration.

Evaluation Phase:

Now that we have setup what we want to be tested and briefed our evaluators we can test the prototype. This first evaluation will allow the users to gain familiarity with the prototype and discover the features we want them to evaluate as well as evaluate the prototype in general. In this stage we let the evaluator familiarise themselves with the app for 30 minutes before we pointed them in the direction of the specific list of heuristics, we wanted evaluated.

Collecting & Filtering Data:

The next phase will be to get the evaluators to record their data (In our provided format of a questionnaire). We have chosen to do this with users to get a more accurate assessment of the evaluator's opinion. Next, we have conducted the initial filtering of the data we have collected from our fellow course students. This required us to average all the results from our survey to produce a mean average and remove any outliers or non-answers. We must make sure that a question has been answered before we include it in the average. This is an important step to ensuring the quality of the feedback we are collecting and its accuracy. As per qualitative data, we read through user's input and assigned code to it and stored them in a table. (example on page 3)

Debriefing & Finalising

This stage will be used to allow the evaluators to discuss their opinions and if they want to, modify their answers to the questionnaire. This will allow for a more rounded opinion and some group feedback. All of this data that has been collected will then be used to improve the design and develop a new prototype where we will do the same set of testing again, but with the same set of heuristics. We have currently repeated this 2 times and have made significant progress in the development of our prototypes according to the feedback we have gathered.

Ten questions were asked to 10 users in the survey. The above chart shows the user's responses to some of the questions. We calculated the average and decided if its equal or below 7 we will work on those areas and find ways to improve. For example, we can assume locating a book in the library is an issue as the average came out to be '4.5'.

Results of Usability Evaluation Methods

Ten questions were asked to 10 users in the survey. The above chart shows the user's responses to some of the questions. We calculated the average and decided if its equal or below 7 we will work on those areas and find ways to improve. For example, we can assume locating a book in the library is an issue as the average came out to be '4.5'.

Heuristic Evaluation Table

Based on user input, we were able to identify the following error and rectify them in the next version of prototype.

Nielsen and Malloch's 10 User Interface Design Heuristics-Evaluation	
Visibility of System Status	The System doesn't indicate the status of the current page. Some of the pages are missing titles. Buttons don't highest when you are on that page. Book button on the home page is very vague and doesn't indicate what it represents.
User Control and Freedom	The system should provide the user to cancel any step taken. Sign up page doesn't have a cancel button. Hidden Logout Button, user found hard to logout of the app as button was provided within the "More"
Match between the system and the real world.	Location of the Search Bar and Home page was pointed out as it was not in a traditional location (Top right-hand side). Use symbols where appropriate: for example, search bar. It's commonly used in most websites.
Consistency and Standards	Different Button style is used within a system. The System breaks external consistency as no help feature is provided within the new system. Book button on the home page is very vague and doesn't indicate what it represents.
Error Prevention	The System automatically renews the books that users have borrowed up to 4 times. This is a potential error, as users might forget to return the book as its auto-renewing and will end up with overdue fee charges.
Flexibility and Efficiency of use,	Only one search option is provided. If the system provides a different category to search by, it will be easier for the user to quickly find the books they are looking for.
Recognition vs recall in the user interface.	Menu bar location was pointed out again, along with the search bar.
Help Users Recognize and recover from error.	When entering invalid input, the system informs the user of the error. For example: accidentally press on the logout button. It straight away logs user out. Pop up window could be added to resolve this issue.
Aesthetic and Minimalist Design	"Welcome Feature" on homepage servers it purposes to some degree, however, takes up the majority of the space. This unit of information competes with more relevant unit of information such as my list, recommended books, etc.
Help and Documentation	Poor: Standard text is provided indicating the floor and shelf number to help the user locate books. Some people find it very hard to locate the book using this method.

Data Source	User input	Code
Questioner	Traditional way of locating book is outdated now.	Book Locating
Observation	"I don't know where is log out button located"	User Interface
Observation	"Button style is not consistent"	User Interface

The Table on the left represent some of the result and how we filtered from the qualitative data that we have collected.