BarAdvisor - An App for Finding the Cheapest Bars

Requirements Document for the Course DAT231

Izabell Arvidsson (izabellarvidsson@live.se),
Moa Berglund (moa.berglund00@hotmail.com),
Erik Eriksson (erik.g.eriksson1@gmail.com),
Oscar Forsyth (oscar.forsyth61@gmail.com),
Simon Larsson (zico.xp.larsson@gmail.com),
Thushini Ranasinghe (thushinirana1998@gmail.com)

September 30, 2022

Change history

Date	Section	Comment	Author
2022-Sep-21	2.2.2	Added feature requirements	Moa Berglund
2022-Sep-21	2.2.1	Added new task description for bar rating	Thushini Ranasinghe
2022-Sep-22	1.5	Added quality grid	Everyone
2022-Sep-26	1	Changed background and goal and scope of project	Oscar Forsyth
2022-Sep-26	2.2.1	Changed/extended work areas	Moa Berglund
2022-Sep-27	-	Slightly changed User Stories, business goals and other small changes	Everyone
2022-Sep-28	-	Added more descriptive figure texts	Erik Eriksson
2022-Sep-28	2.4	Added the prioritization data + text	Erik Eriksson
2022-Sep-28	1.3.3	Added goal domain tracing	Oscar, Simon
2022-Sep-29	2.3	Added QRs and Quality models	Simon, Erik
2022-Sep-29	1.3.1	Edited Business goals	Moa Berglund
2022-Sep-29	3.2	Added mockups for BarAdvisor	Thushini Ranasinghe

Contents

1 High-level description

This document outlines the requirements, scope, and project for the BarAdvisor App, which aims to assist students in locating affordable beverages within Gothenburg.

1.1 Background

The price of any given drink at a bar directly affects the likelihood of a student choosing or returning to that specific bar (Refer questionnaire 4.1). Current solutions for students to find an affordable drink at a bar are very limited in the city of Gothenburg. These solutions consist of article recommendations, scrolling through Google reviews and Tripadvisor type applications. However, these alternatives provide lackluster or even absent solutions to finding the specific cheapest price at any given bar. Most of these alternatives focus on providing reviews of bars while we want to create an application that focuses on providing the user with accurate drink prices. Many bars want to increase their student customer base and choosing to advertise on the application could do this. The Student Union wants to improve the student experience as well as have an income.

1.2 Goal and scope

The application, BarAdvisor, will help students navigate through bar prices and locations. It will do this by providing a search and filtering option. Monetization of the application will be done through sponsored advertisements visible on the home page. The goal of the application is to make drink prices at bars, in the city of Gothenburg, available to anyone, especially students. In doing this we hope to improve the quality of life and improve the overall university experience for students in Gothenburg.

The scope of the application consists of listing drink prices at bars, general bar information and possibly promoting them. It does not include user directions to bars nor any type of booking system.

The context diagram below shows the system's relationships and how they function.

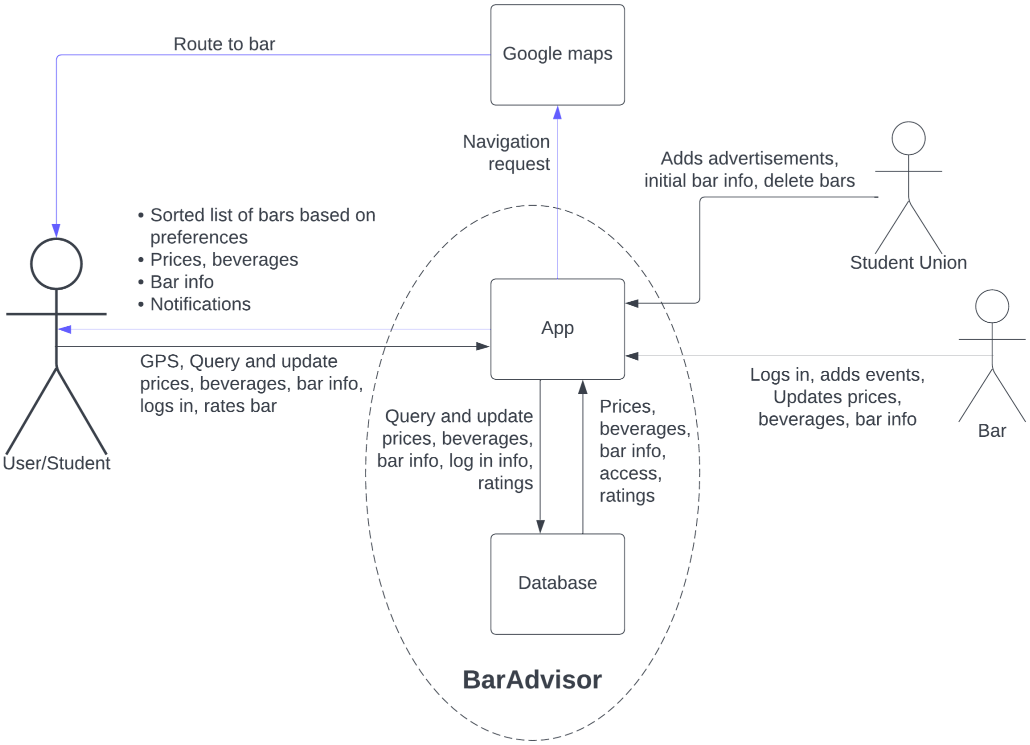
Google maps
Route to bar
Navigation request
Adds advertisements, initial bar info, delete bars
Student Union
• Sorted list of bars based on preferences
• Prices, beverages
• Bar info
• Notifications
App
GPS, Query and update prices, beverages, bar info, logs in, rates bar
Logs in, adds events, Updates prices, beverages, bar info
Bar
User/Student
Query and update prices, beverages, bar info, log in info, ratings
Prices, beverages, bar info, access, ratings
Database
BarAdvisor

Figure 1: Context diagram for BarAdvisor, showing an overview of how the internal system, outlined by the dotted line, interacts with external entities. In the diagram, boxes define high-level systems, while characters represent stakeholders, with the arrows in between signifying data transactions.

1.3 Business case and stakeholder map

In this section, the stakeholders and users are explained, as well as high level reasons (business goals) for developing BarAdvisor.

1.3.1 Business goals

BarAdvisor aims to reach these business goals:

Business Goal 1: Facilitate bar search for students. BarAdvisor should provide an easy and clear way to find affordable bars in the vicinity. This is solved by BarAdvisor offering a platform where all bars in the area are listed relative/in accordance to search and filter inputs.

Business Goal 2: Enrich student life. This is done by BarAdvisor facilitating to make budget-friendly choices when wanting to visit a bar, so that students optimizes their budget. BarAdvisor also aims to enrich student life by promoting campus life through the affordable beverages, fun activities and general proximity of student bars. This is done by ranking bars by price, including showcasing available events and offering an integrated map view with distances displayed.

Business Goal 3: Support bars in reaching out to students. This is done by distributing the app where the bars are shown to the students. This is also done by allowing bars to buy advertisements in BarAdvisor to promote different events and offers suitable for students.

1.3.2 Stakeholders

This section provides an overview of our app's stakeholders.

Name	Relationship	Representative	Power/Sentiment
Students	User	Krystyna	Low, will use
Chalmers student union	Funder	Viktor Kjellson	High, will maintain, distribute, and fund
Bar owner	Client	Thomas Jordansson Lawner	Mid, will support project

Table 1: Stakeholder map for BarAdvisor

Stakeholder 1: Student. Students often are on a strict budget, but many still value an active student life with events and bar visits with friends. In BarAdvisor students are the users.

Stakeholder 2: Chalmers student union. The student union is concerned with accommodating a thriving student life for the students. They are a center point for campus life and have contact with the committees arranging the student bars. They keep in contact with students mainly through social media channels. In BarAdvisor the student union would host the application to distribute a solution to help the students easily locate bars suitable for their budget.

Stakeholder 3: Bar owner. The client wants to promote their bar and reach out to more potential guests. The bars concerned in this project are both on campus and "ordinary" bars outside the student clientele. The bar owners buy advertisements to promote their bars. With BarAdvisor, the clients get a well founded platform to reach out to the student audience.

1.3.3 Goal Domain Tracing

The goal domain tracing in Table 2 displays the relationships between business goals and core functionality.

User Story	BG 1	BG 2	BG 3
(US1) Search	x		
(US2) Filter	x		
(US3) Display	x		
(US4) Locate	x		
(US5) Navigate	x		
(US6) Price update	x		
(US8) Rate	x		
(US10) Add Advertisements		x	
(US11) Advertise			x
(US13) Editing privileges			x
(US14) Promotions			x

Table 2: Goal domain tracing

1.4 Core functionality

This section displays the core functionality of BarAdvisor. User stories are used to help see the value of the product and give a better understanding of why the stakeholders want certain functionality.

US1: As a student I want the app to have a search function so that I can find information about a certain bar.

US2: As a student, I want to be able to filter out bars based on specific beverages so that I can go to bar that has my preferred beverage.

US3: As a student, I want the price of the beverage displayed so that I can find the cheapest one.

US4: As a student, I want the app to have some sort of map so that I can see the bars closest to me.

US5: As a student, I want to be directed to my phone's navigation app so that I can get a route to the bar I've chosen.

US6: As a student, I want to be able to suggest new prices so that outdated prices are corrected.

US8: As a student I want to be able to leave ratings of bars so that I can share my thoughts regarding the bars.

US10: As the Student Union we want to be able to add advertisements to the app so that we get profit.

US11: As a bar owner I want to be able to make requests for adding ads in the app so that my bar gets seen by a larger number of people.

US13: As a bar owner, I want to have an admin account so that I can update/change the information of my bar.

US14: As a bar owner, I want to show what activities my bar offers as well as upcoming events, so that I can attract more guests.

1.5 Performance Requirements, Specific Quality Requirements, Constraints

The non-functional requirements for BarAdvisor are divided into three large categories: *operation*, *revision* and *transition*, each with underlying quality factors. Of these categories, the qualities pertaining to *operation* and *revision* are all relevant in measuring the success of BarAdvisor, while the majority of *transition* related qualities do not align with any of the business goals.

In regard to *operation*, all business goal can be mapped to core functionality as can be seen in Table 2, and thus require quality requirements to measure fulfillment. Similarly, *revision* considers necessary factors to ensure that BarAdvisor fulfills these business goals long term. In contrast, *transition* related qualities such as transferring the program between systems (portability) or reusing it as part of other applications (reusability) are not factors that the stakeholders are concerned with.

The complete prioritization of quality factors, along with explanations for certain choices, can be viewed in Table 3. Each quality factor is based on the definitions given in McCall and Matsumoto, with the exception of ”Installability” which was not part of the original quality factors, but added to account for non-trivial distribution requirements.

	Critical	Important	As usual	Unimportant	Ignore
Operation					
Integrity/security			x		
Correctness	1				
Reliability/availability			x		
Usability			x		
Efficiency			x		
Revision					
Maintainability			x		
Testability			x		
Flexibility		2			
Transition					
Portability					x
Interoperability					x
Reusability					x
Installability			x		

Table 3: Quality grid for BarAdvisor, showing how important each quality factor is. Concerns for the more important factors are given a number and can be read below.

Concerns:

1. Correctness

Fulfilling users' mission objectives is essential for BarAdvisor, as it is the driving force behind stakeholder interest. With underprioritized *correctness*, core functionality would be comparatively worse, which would hurt business goal satisfaction, reducing stakeholder interest.

An example of *correctness* could be: "The search results must be in line with the inputted filters." This would be derived from BG1, where students' primary mission objective is to find a suitable bar, but if the filters are poor, the bar search is instead hindered.

2. Flexibility

As a stakeholder with high power, the student union, has expressed limited willingness to manage BarAdvisor's day-to-day operations. Therefore, the effort required to maintain and adapt the app to changing market demands should be minimal.

Integrity/security

QR 1: The product must follow OWASP Mobile Application Security Verification Standard-L1 (MASVS-L1) for application security and integrity.

For the following sub-requirements of OWASP MASVS-L1, more details are provided on how they must be achieved:

MSTG-ARCH-1: The sensitive data the product shall handle are email addresses and passwords.

MSTG-STORAGE-12: There must be a way to show a privacy policy on the login screen as well as when logged in.

MSTG-AUTH-5: The password policy must be the following:

- The password must be at least 15 characters long.
- The password must not exceed 100 characters.
- Any combination of letters, numbers, and symbols are allowed.

Correctness

QR 2: Attractive aspects of the bar must be displayable through at least 5 different ways.

QR 3: When searching, the app must load the cheapest bars in ___ seconds. (Customer expects 2 seconds.)

QR 4: Changes to app content must be updated for all users in a reasonable time frame:

- x minutes for advertisements
- y seconds for minor bar information such as ratings, description, and pictures
- Upon restart of the app for major bar changes such as addition/removal of bars

QR 5: The search results must be in line with the inputted filters.

QR 6: The app must show cheap bars before expensive ones.

QR 7: The advertisement placement must not be intrusive to avoid hindering the student user from reaching their goal.

QR 8: The filtering process must not exceed 30 seconds for the great majority of new users when they have a clear goal in mind.

Reliability/availability

QR 9: If the servers are down, the product must still be able to search, filter, browse, and show the map view.

QR 10: The product's map must come from a trusted third party and be updated once a year.

QR 11: The bar info displayed in the product must be reliable.

Usability

QR 12: The product interface shall be familiar to the great majority of users.

QR 13: The product must make it easy to update data for users with the bar role.

QR 14: All text in the product must be easily comprehensible in regard to wording and font.

QR 15: The process of adding advertisement in the app takes less than ___ minutes. (Customer expects 10 minutes)

Efficiency

QR 16: Using the product must consume less than or equal amount of battery to that of similar products.

QR 17: Using the device GPS when looking at the map view must not noticeably increase battery consumption compared to not using the GPS.

QR 18: The app must be able to support 5000 concurrent users without noticeable performance drop.

Maintainability

QR 19: Comprehensive documentation for new patches must be available at ___.

QR 20: A RAD in addition to code must be available for authorized parties at ___.

QR 21: The app must have a simple process for bug reporting.

Flexibility

QR 22: Major modifications to the system must be supported through monthly patches.

QR 23: The app must work despite running on an older version.

Installability

QR 24: The product must be available to download on Google Play for Android devices and The App Store for devices running iOS.

QR 25: The minimum operating system version the app must be available for must include 70% of all users' devices for that operating system.

2 User Requirements Specification

In this section, the user requirements are provided. It begins with providing the data requirements, followed by the functional requirements, and then stating the quality and performance requirements. Lastly, this section investigates in prioritization of the requirements.

2.1 Data requirements

This section describes the data requirements identified by us for our application. The data requirements are listed down below and furthermore we have included an ER diagram to show which data are kept in the database and how they are related.

DR1: The database part of the product shall follow the ER-model in figure 2 for storing data.

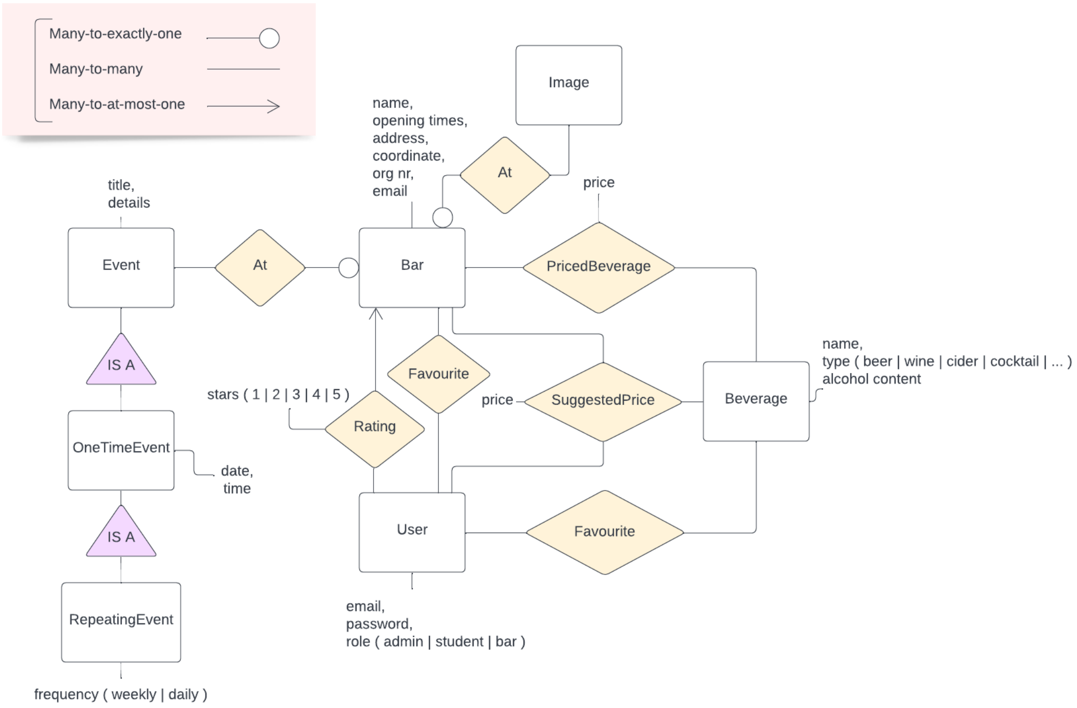

Figure 2: ER-model for BarAdvisor showing how the entities (white rectangles), with their attributes, can be modelled through relationships (shown with purple and orange shapes).

DR2: The product shall support adding favourite bars and beverages without being logged in to a specific user, these will be called local favourites.

DR3: The product shall remove local favourites when logged in and add them back when logged out.

DR4: The device must be able to cache the latest fetched data so that it can be shown in case the database is unreachable.

2.2 Functional requirements

This section lists the functional requirements. It begins with task descriptions to make some critical tasks clear, and continues with stating some less complex feature requirements.

2.2.1 Task descriptions

All tasks below are connected to a work area to place the tasks in the right context. This section presents each work area followed by its connected tasks descriptions.

Work area: 1. Finding the cheapest bar
Most often, the user (a student) is part of a group where each has their own beverage preference but everyone has similar budget constraints. Some may already know of a few bars, but if prodded about details, they are unable to give a fair and factual description. They are mainly searching for nearby bars during the evening/night (peak hours) on their smartphone. Because of this, walking and public transport are preferred over cars.

Another common user scenario is a student using the application to plan ahead for a bar visit. Could be at various places, at home in couch or on the bus for example.

Users: Experienced smartphone users that are somewhat familiar with the local area. Possibly intoxicated. Main concerns: price, supply, and proximity.

FR1: The product shall support task 1.1 as seen in table 4.

FR2: The product shall support task 1.2 as seen in table 5.

FR3: The product shall support task 1.3 as seen in table 6.

Task:	1.1 Search
Purpose:	To find a cheap bar with desired beverages
Trigger:	Student wants to visit a bar
Frequency:	Almost every time the app is used
Critical:	Wants to filter for more than one drink
Sub-tasks:	
1:	Filter for the desired drinks
2:	Excluding certain types of the desired drinks
3:	Including allergies
Variants:	
1a:	The desired drink is not in the filter section
3a:	Pre-entered allergies

Table 4: Task description 1.1 Search

Task:	1.2 Surfing the app
Purpose:	Looking for bar information
Trigger:	Wants to plan going to a bar or view different offers
Frequency:	Every time student opens up the app
Critical:	
Sub-tasks:	
1:	Comparing prices of drinks
2:	Going through the gallery, menus, and other info of bars
Variants:	
2a:	Bar owner hasn't provided the necessary information

Table 5: Task description 1.2 Surfing the app

Task:	1.3 Locate bar
Purpose:	Get directions to a bar, and navigate to a bar.
Trigger:	User has found the desired bar on the app's map
Frequency:	1 time / bar search
Critical:	
Sub-tasks:	
1:	Gets directed to the phone's navigation app
2:	Approve/choose route to bar
3:	Navigate to bar
Variants:	
2a:	User rejects suggested route
2b:	User selects different route
3a:	User cancels plan
3b:	Route to the bar cannot be followed

Table 6: Task description 1.3 Locate bar

Work area: 2. Student Union office
The student union is hosting the application by managing advertisements, requests from bar owners, and user accounts, by being logged in to an admin account. Only one user from the student union will use their admin account at a time. The user will probably sometimes interact with this app and a mail app back and forth, since they will enter info from their mailbox (bar advertisement info) into the app.

Users: Experienced smartphone users. The student union has much on their plate, so the user could be stressed.

FR4: The product shall support task 2.1 as seen in table 7.

Task:	2.1 Add advertisement
Purpose:	Add/update an advertisement in the app
Trigger:	Bar owner wants to buy an advertisement
Frequency:	Average 1-2 ad/bar/month
Critical:	No suitable advertisement template exists
Sub-tasks:	
1:	Find a suitable advertisement template
2:	Fill in the data and publish/schedule
3:	Sending an invoice to bar owner
Variants:	
1a:	Need to make a new advertisement template
2a:	Repeated advertisement
3a:	Send to different receiving channels

Table 7: Task description 2.1 Add advertisement

Work area: 3. Student interactions
Location: at bar or recently visited this bar. If the user is at the bar, the environment is loud, possibly with friends interrupting with conversations. Most often this situation will occur at the evening/night. If the user recently visited the bar, he might be on his way home, or already at home, after a night out. Digital device: smartphone.

Users: Students. Experienced smartphone users. Possibly intoxicated or tired.

FR5: The product shall support task 3.1 as seen in table 8.

FR6: The product shall support task 3.2 as seen in table 9.

Task:	3.1 Students edit prices
Purpose:	Students should be able to edit the prices of drinks at bars in case the prices has changed
Trigger:	Student notices that price is incorrect
Frequency:	On average one time every few weeks to a few months/user
Critical:	Max 5 suggestions / day / user
Sub-tasks:	
1:	Student must login
2:	Navigate to the bar and the certain drink
3:	Leave a price suggestion
Variants:	
1a:	Student must register
2a:	The drink does not exist
3a:	Vote on an already existing suggestion

Table 8: Task description 3.1 Student edits prices

Task:	3.2 Students add bar ratings
Purpose:	Students should be allowed to provide ratings for bars they've been to or are familiar with
Trigger:	Student wants to rate a bar
Frequency:	On average, once every few weeks to months per user.
Critical:	
Sub-tasks:	
1:	Student must login
2:	Navigate to the bar
3:	Rate the bar
Variants:	
1a:	Student must register
2a:	The bar does not exist
3a:	Cancel rating

Table 9: Task description 3.2 Student add bar ratings

Work area: 4. Bar

Bar owner/ responsible team can manage the bar information, and keep contact with the student union to plan ahead for certain advertisements. This will probably be conducted at the bar on closed hours, or at an office. This could also occur in a more stressful environment, if a guests reports that something is wrong directly to the bar staff during opening hours. In that case, the staff might want to correct the information on the fly.

Users: Novice to somewhat experienced smartphone users.

FR7: The product shall support task 4.1 as seen in table 10.

Task:	4.1 Bar owner edits bar data
Purpose:	The bar owner should be able to edit the data for their bar (such as opening hours, prices, description, events, etc.)
Trigger:	A bar owner has made change to e.g. the assortment
Frequency:	On average a few times a month per bar
Critical:	
Sub-tasks:	
1:	Log in with bar account
2:	Find item(s) that need updated information
3:	Write in new data
4:	Save new data
Variants:	
1a:	Already logged in
1b:	Logged in to a non-admin account
2a:	Add new item(s)
3a:	Remove item(s)
4a:	Cancel changes

Table 10: Task description 4.1 Bar owner edits bar data

2.2.2 Feature requirements

This section covers a few more functional requirements, but these do not require a full task description to be understood. Therefore, these are stated in a simpler way, as feature requirements.

FR9: The application shall begin with an age verification popup to ensure that the user is of legal drinking age, followed by a prompt stating ”Don’t drink and drive.”.

FR10: The application shall support disabilities aligned with the Web Content Accessibility Guidelines (WCAG) 2.1.

FR11: The application shall be able to show a map where all bars are marked. It shall be possible to search for different areas in the map to show the bars in that area.

FR12: The application shall allow the admin account to delete accounts, if misused or if students/bars no longer are valid.

2.3 Detailed Performance Requirements, Specific Quality Requirements, Constraints

After prioritizing the most important quality attributes listed in Section 1.5 this section describes the selected attributes in more detail.

For quality requirements: QR3, QR4, QR7, QR8; QR12, QR13, QR15; QR17, further details are provided in the quality models in Figures 3, 4 and 5.

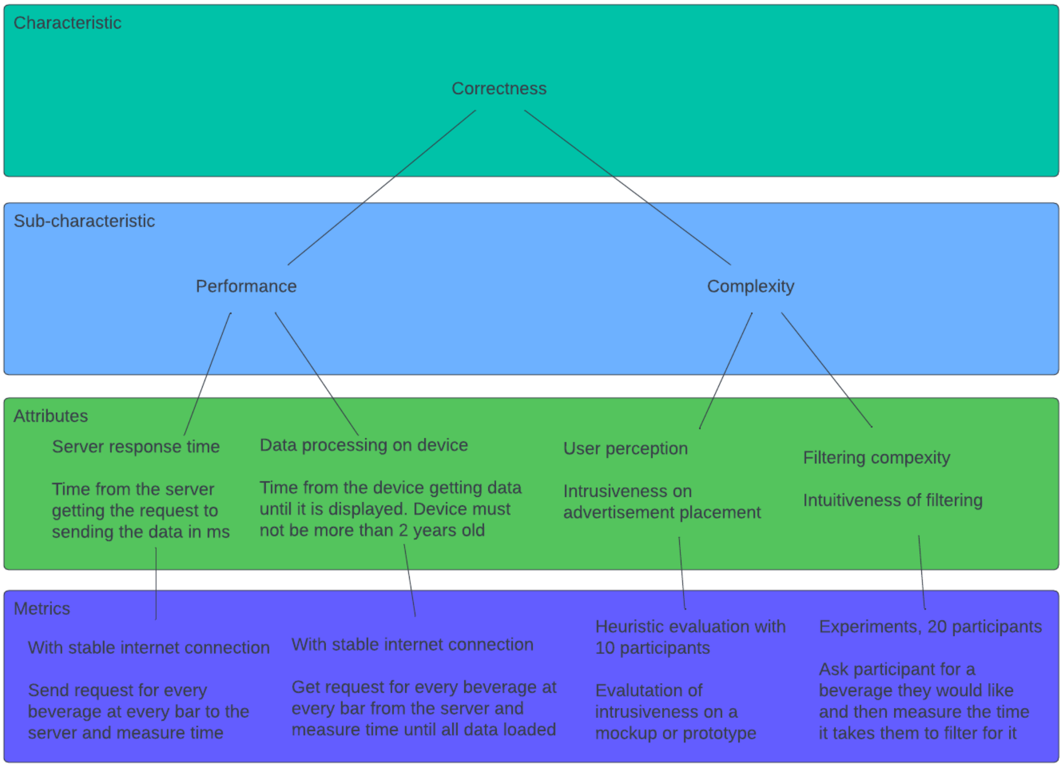
Characteristic
Correctness
Sub-characteristic
Performance
Complexity
Attributes
Server response time
Time from the server getting the request to sending the data in ms
Data processing on device
Time from the device getting data until it is displayed. Device must not be more than 2 years old
User perception
Intrusiveness on advertisement placement
Filtering compexity
Intuitiveness of filtering
Metrics
With stable internet connection
Send request for every beverage at every bar to the server and measure time
With stable internet connection
Get request for every beverage at every bar from the server and measure time until all data loaded
Heuristic evaluation with 10 participants
Evalutation of intrusiveness on a mockup or prototype
Experiments, 20 participants
Ask participant for a beverage they would like and then measure the time it takes them to filter for it

Figure 3: Quality model for correctness, showing details on how QR3, QR7, and QR8 are to be evaluated. From left: QR3, QR3, QR7, QR8.

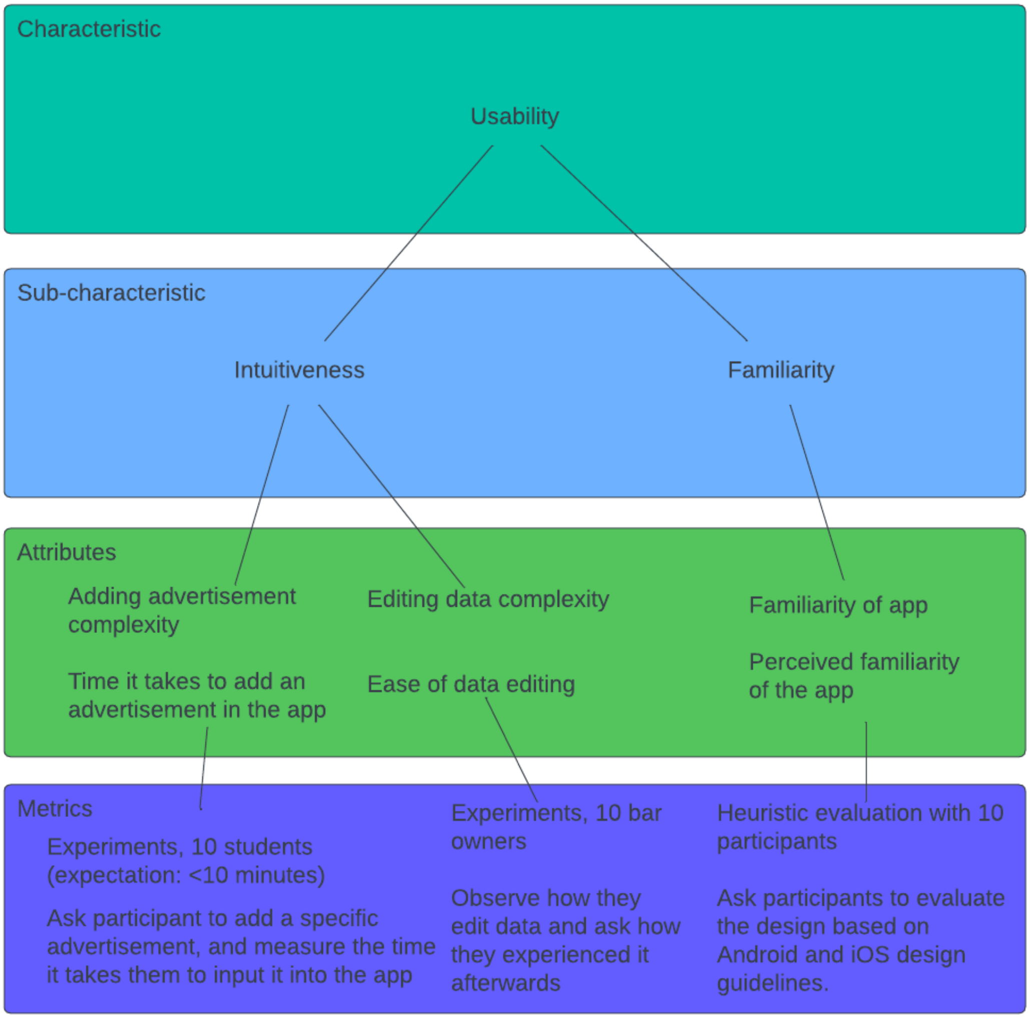
Characteristic
Usability
Sub-characteristic
Intuitiveness
Familiarity
Attributes
Adding advertisement complexity
Time it takes to add an advertisement in the app
Editing data complexity
Ease of data editing
Familiarity of app
Perceived familiarity of the app
Metrics
Experiments, 10 students (expectation: <10 minutes)
Ask participant to add a specific advertisement, and measure the time it takes them to input it into the app
Experiments, 10 bar owners
Observe how they edit data and ask how they experienced it afterwards
Heuristic evaluation with 10 participants
Ask participants to evaluate the design based on Android and iOS design guidelines.

Figure 4: Quality model for correctness, showing details on how QR12, QR13, and QR15 are to be evaluated. From left: QR15, QR13, QR12.

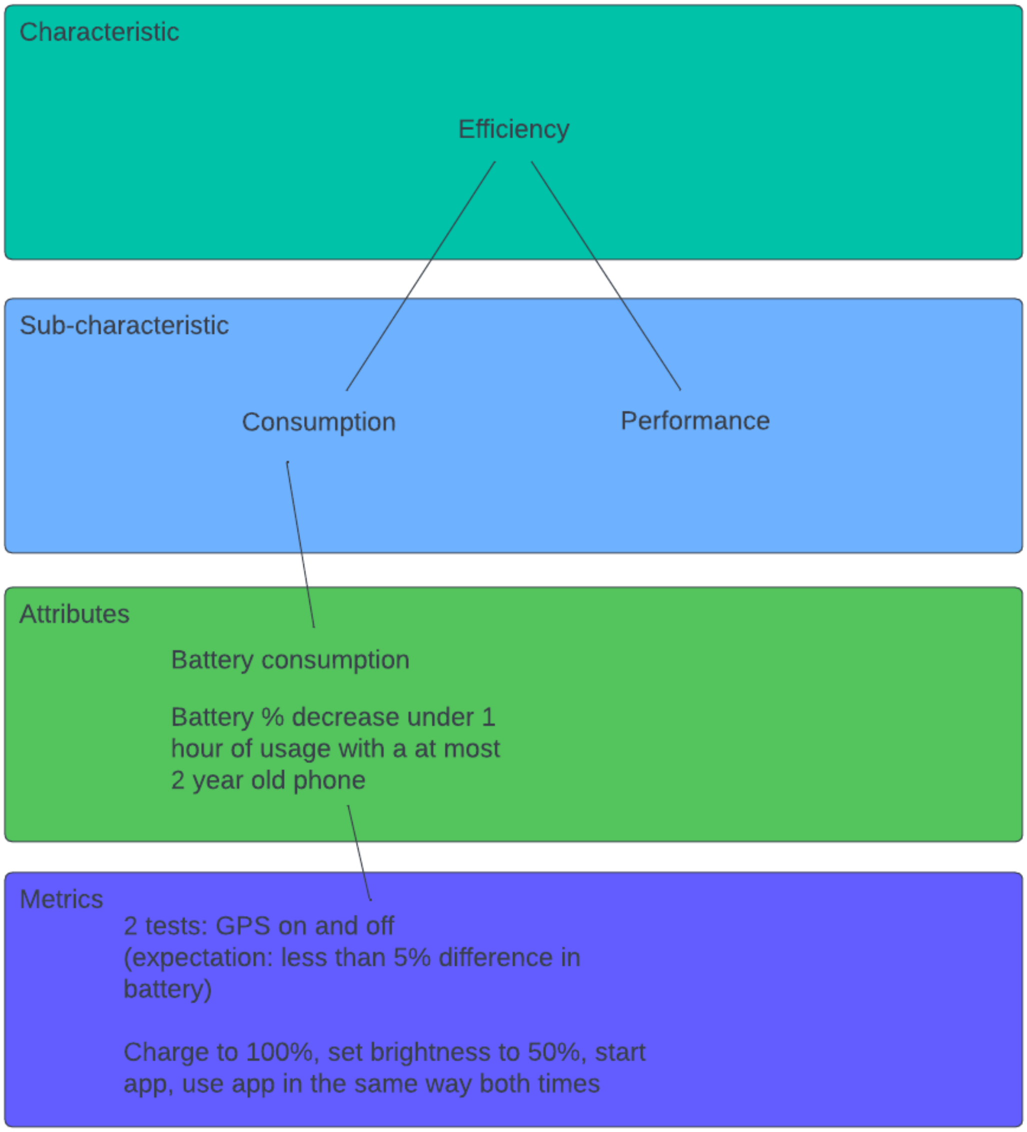
Characteristic
Efficiency
Sub-characteristic
Consumption
Performance
Attributes
Battery consumption
Battery % decrease under 1 hour of usage with a at most 2 year old phone
Metrics
2 tests: GPS on and off (expectation: less than 5% difference in battery)
Charge to 100%, set brightness to 50%, start app, use app in the same way both times

Figure 5: Quality model for efficiency, showing details on how QR17 is to be evaluated. From left: QR17.

2.4 Proposed prioritization

Two techniques were used to prioritize requirements, $100 test and the top ten requirements techniques.(See appendix 4.2) Below are compiled versions of the prioritizations we are proposing.

Table 11 shows the compiled priority for the functional requirements (which includes the data requirements) using the $100 test with $100 for each stakeholder. It makes a lot of sense that the tasks that the app must support, FR1 to FR7, are of the highest priority with the exception of FR5 which is a bit lower. Despite DR1 being priority 9 in the compiled data, it must be of higher prioritization in reality since that is the foundation everything else is built upon.

Priority	Sum of $	Requirement
1	51	FR2
2	37	FR1
3	34	FR3
4	31	FR4
5	27	FR7
6	25	FR6
7	21	FR10
8	16	FR11,FR5
9	13	DR1
10	8	FR9
11	7	DR2,DR3,FR12

Table 11: Compiled data of the $100 technique for the functional requirements

Table 12 shows the compiled priority for the quality requirements using the $100 test but with $200 for each stakeholder this time. The reason is simply because there are more of the quality requirements compared to the others. The data solidifies that correctness is the most important aspect of the project.

Priority	Sum of $	Requirement
1	62	QR5
2	60	QR1
3	51	QR2
4	37	QR11
5	36	QR6
6	35	QR13
7	34	QR8
8	33	QR3
9	26	QR24,QR7
10	23	QR12
11	21	QR18
12	18	QR15
13	17	QR23,QR14
14	16	QR25,QR9
15	15	QR16,QR4
16	12	QR10
17	9	QR20
18	8	QR19
19	7	QR21
20	3	QR22,QR17

Table 12: Compiled data of the $100 technique for the quality requirements

Table 13 shows two sets of requirements; requirements that were in any of the stakeholders' top ten list and requirements that were not in any of them.

Top Ten Requirements	Other Requirements
FR1	FR9
FR2	FR10
FR3	FR11
FR4	FR12
FR5	DR1
FR6	DR2
FR7	DR3
QR1	QR4
QR2	QR7
QR3	QR9
QR5	QR10
QR6	QR14
QR8	QR16
QR11	QR17
QR12	QR19
QR13	QR20
QR15	QR21
QR18	QR22
QR24	QR23
	QR25

Table 13: Compiled data of the Top Ten Requirements technique

2.4.1 Next release

The next release of BarAdvisor must support FR1 through FR11 and DR1, as well as be accepted in evaluation of QR1 through QR16, QR24, and QR25.

2.4.2 Second release

The second release of BarAdvisor must also support FR12, DR1, and DR2, as well as be accepted in evaluation of all QRs.

3 System Requirements

Will consider in R3.

3.1 System requirements

3.2 UI Prototype

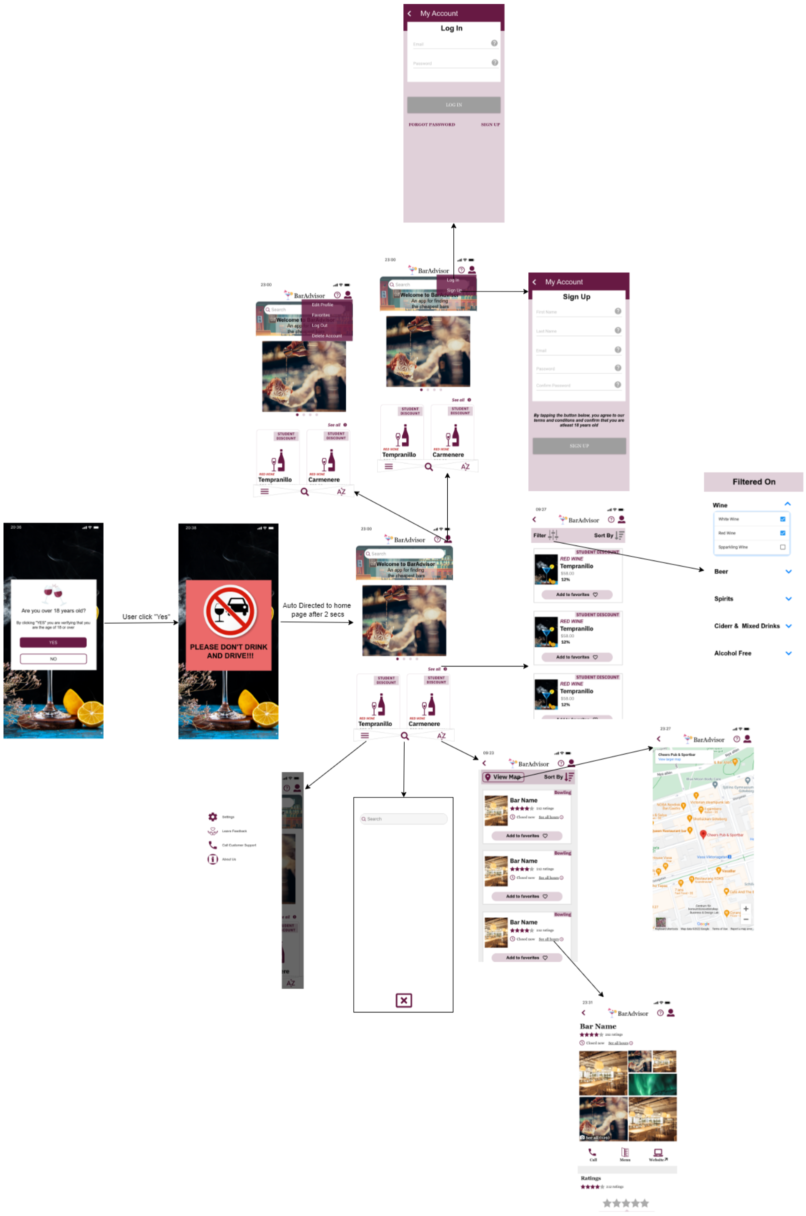

Figure 6: Mockups for BarAdvisor.

3.3 Detailed Data Requirements

3.4 Acceptance Tests (Optional)

4 Appendix

4.1 Questionnaire

See this link for the full questionnaire.

4.2 Prioritization

Refer this link for prioritization.