Software Development Plan

MallTrails

Prepared By:
Emirhan Toprak, Kian Ansarinejad, Sabahaddin Ispiroglu,Murat Göçmen,Eren Kösen

17/05/2023

TABLE OF CONTENTS

1.1) Project Description

This project aims to solve the difficulties shoppers face in navigating large, complex shopping malls using traditional methods such as paper maps or asking for directions. These methods can be time-consuming, confusing, and frustrating for shoppers. To address these issues, we propose developing a mall navigation app that utilizes indoor mapping technology to guide users through the mall efficiently and effectively. The app will provide turn-by-turn directions, real-time location tracking, and other helpful information to simplify the shopping experience for users and provide them with a more enjoyable and stress-free experience.

1.2) Objectives

The objectives of the mall navigation app are as follows:

- Finding Stores: Help users find the stores they are looking for in a mall.
- Indoor Navigation: Use indoor mapping technology to help users navigate through a mall's complex layout.
- Saving Time: Provide users with the most efficient route to their destination to help them save time.
- Promoting Deals and Sales: Promote deals and sales from stores within the mall to help users discover new stores and products while also saving money.
- Personalized Recommendations and Discounted Store Navigation: Utilize data analysis to offer personalized product recommendations based on the user's previous shopping history and favorite products. The app should also help users find stores with better discounts, providing valuable insights and recommendations to mall retailers.
- Enhancing the Shopping Experience: Provide users with additional information about the mall, such as store hours, restaurant recommendations, and parking information to enhance their overall shopping experience.

2. Requirements

2.1) Functional Requirements

- Map and Store Location Search: The app should provide a map of each mall and allow users to search for the location of each store within the mall.
- Real-time Turn-by-turn Directions: The app should provide users with real-time turn-by-turn directions with alternative routes and adjustments for obstacles or closures.
- Store Hours: The app should display updated information about the working hours of stores within each shopping mall.

- Store and Product Category Search: The app should have a search panel so that users can search for stores and product categories within the mall.
- Personalized Recommendations: The mall navigation app must track user preferences and generate personalized recommendations based on their favorite product categories.
- Notifications: The mall navigation app must notify users of new store openings and permanent closures within each shopping mall.
- Nearby shopping malls: The mall navigation app must identify nearby shopping malls based on the user's location and make suggestions for mall options.
- Retailer Analytics: The app must provide retailers with users' popular brands and product categories data to customize their offerings and promotions.

2.2) Non-functional Requirements:

- Language: The application must be developed using Java programming language, version 11 or later.
- Usability: The app should be user-friendly, accessible, and visually appealing.
- Performance: The app should be fast (with a response time of 0.2 seconds or less), reliable, and able to handle a high volume of users, such as 10,000 concurrent users during peak hours, with the highest response time.
- Security and Safety: The app should be safe to use, and all sensitive user information should be encrypted using industry-standard encryption methods such as AES-256. Possible leaks of user personal information should be prevented through regular security audits and updates.

- Platform: The application should work on different platforms (iOS 13 and later & Android 10 and later).
- Efficiency and Lightweight: The app should be memory-efficient, with a small download size of no more than 50MB, for easy installation on users' devices.

3. Stakeholders

- Shoppers /End users: Shoppers are the primary stakeholders in the mall navigation app project as they will be the primary users of the app. A successful project can lead to a positive impact on shoppers, as it can provide a user-friendly tool to navigate the mall, leading to increased convenience and satisfaction. This results in higher customer loyalty and increased revenue for retailers. However, if the project fails, shoppers may experience frustration and inconvenience due to the app's poor functionality, leading to a negative impact on their shopping experience. This can result in a decrease in customer loyalty and potential revenue loss for retailers.

- Project Managers: Project Managers are key stakeholders in any project, including our project. They are responsible for leading the project team, ensuring that the project is delivered on time, within budget, and to the required quality standards. A successful project can enhance their reputation, increase their job opportunities, and demonstrate their ability to deliver complex projects. Conversely, a failed project

can damage their reputation, limit their career progression, and reduce their opportunities for future projects.

- Development Team: The Development Team is responsible for designing, developing, testing, and implementing the app project. A successful project can lead to increased career and business opportunities, job satisfaction, and motivation. However, Project Failure can damage their reputation, decrease morale, and increase stress, affecting their personal and professional lives.

- Mall Management Mall Management is a vital stakeholder in our project. They are responsible for ensuring that the mall runs smoothly and meets the needs of its customers. The success of the project can have a positive impact on Mall Management, such as increased foot traffic, sales, and customer satisfaction. Also, a successful project can attract new businesses, and differentiate them from their competitors. On the other hand, failure can have negative consequences for Mall Management, such as decreased sales, and customer satisfaction, leading to a loss of revenue and businesses. Moreover, failure can damage their reputation, resulting in a negative impact on future business prospects.

- Retailers: Retailers are responsible for operating stores and ensuring customer satisfaction. Success of the mall navigation app can increase sales and improve customer loyalty. It can also provide marketing opportunities and customer data for analyzing behavior and preferences. Failure can decrease sales, and customer engagement, leading to lost opportunities for marketing and data collection and ultimately leading to lost profits.

- Marketing and Advertising Companies: The mall navigation app can provide marketing and advertising companies with the potential for location-based advertising. Project Success can result in increased revenue from targeted ads, leading to higher sales and profits for both Retailers and advertisers. Failure can lead to lost revenue from location-based advertising, decreased customer engagement, and ultimately lower revenue of marketing companies.

- Data Providers: Data Providers can be stakeholders in the development of our project. They may provide the data for mapping, real-time mall traffic information, and other essential services. A successful project can lead to increased demand for their data, resulting in potential new business opportunities and higher revenue. Conversely, if the project fails, it may negatively impact their reputation and potential future business prospects.

- Investors: Investors are significant stakeholders in the mall navigation app project, as they are responsible for funding the project. Successful project can lead to increased revenue and profits, resulting in a positive return on investment for investors. Additionally, a successful project can enhance their reputation and attract potential investors for future projects. On the other hand, failure can lead to a loss of investment and potentially damage the investor's reputation.

- Competitors: Competitors can be considered as stakeholders in the mall navigation app project, as the success or failure of the project can have a significant impact on their market position. If the project is successful, it can lead to increased competition, as other businesses may want to develop similar apps to keep up with the market trends. also, competitors may face challenges in maintaining their market share, as the MallTrails can attract customers away from their app and business. However, if the project fails, it can have a positive impact on competitors because it could reduce the competition in the market. Project failure can also provide opportunities for competitors to gain a competitive advantage, as they can improve their own navigation apps or offer alternative services to attract customers.

- Government: The government can also be considered a stakeholder in our project, as it can have an impact on the local economy and community. A successful project can increase the number of visitors to the mall and contribute to the growth of the local economy, leading to increased tax revenue for the government. It can also enhance the government's reputation for promoting innovation and supporting businesses in the community.
  However, if the project fails, it can have a negative impact on the local economy and community, potentially leading to a decrease in tax revenue and business opportunities. The government may also face criticism for investing in a failed Project.

4.1) Project Staffing: Software Engineering Roles

The success of any software development project largely depends on the skills, experience, and dedication of the team members involved. For our mall navigation app development project, we will be following the Scrum agile methodology, which requires a well-structured team with specific roles and responsibilities.
We have identified the following key roles and their respective responsibilities for our project:

- Product Owner: This role will be responsible for managing the product backlog, defining the project vision, and ensuring that the development team is working on the most valuable features. The product owner will work closely with stakeholders and the development team to prioritize the product backlog and make decisions that drive the project towards success.

- Scrum Master: The Scrum Master will be responsible for ensuring that the Scrum methodology is followed, facilitating all Scrum events, and removing any obstacles that may hinder the team's progress. The Scrum Master will also coach the team on agile principles and best practices and will act as a liaison between the team and stakeholders.

- Development Team: Our development team will consist of software engineers and a UI/UX designer. They will be responsible for developing the software, testing it, and delivering it according to the product owner's requirements. They will work collaboratively and be responsible for self-organizing and managing their work.

- Project Manager: The project manager will be responsible for overseeing the project, managing the schedule, and ensuring that it is delivered within budget and on time. They will work closely with the product owner to manage the project scope and will be responsible for communicating the project status to stakeholders.

We believe that this team structure will allow us to deliver a high-quality product on time and within budget. Each team member's role and responsibilities have been clearly defined, ensuring that everyone knows what is expected of them. The team will work collaboratively using the Scrum methodology to ensure that the project is delivered according to the product owner's vision and requirements.

5. Software Process Model

5.1.1) NECESSARY NEEDS FROM THE ORGANIZATIONAL PROCESS:

- **Continuous feedback and iteration**: The development team needs to receive continuous feedback from stakeholders and end-users to ensure the app meets their needs.
- **Continuous improvement**: App should always be improving to meet user needs and stay competitive. Process model that is used for developing app should emphasize continuous improvement through regular adjustments to the product requirements.

- **Easy maintenance**: Designing the app with easy maintenance in mind is essential for its long-term success. This includes providing a user-friendly system for reporting issues, making updates easy to implement, and ensuring that the app is easily maintainable.

- **User-focused approach in the development of the mall navigation app**: Using user stories for system specification is a necessary need in development process. By creating user stories, the development team can better understand the needs and desires of the app's users, which can guide the development of the app's features and functionalities.

- **Risk management**: Developing a mall navigation app poses several risks, such as technical issues, changing market trends, and security concerns. Risk management is necessary to identify, assess, and mitigate potential risks regularly. Mitigation strategies can include avoiding the risk, reducing its likelihood or impact, transferring it to a third party, or accepting the risk with a plan to manage it.
- **User engagement testing** allows organizations to gather feedback from users and make adjustments to the product based on that feedback. This helps to ensure that the product meets the needs and expectations of its intended audience.

- To ensure that the mall navigation app is developed within the given timeframe, the development team needs to **prioritize features and plan tasks**. Prioritization involves identifying and ranking features according to their importance.

- **Incremental delivery**: An effective mall navigation app may require incremental delivery of features to end-users.

5.1.2) UNNECESSARY NEEDS FROM THE ORGANIZATIONAL PROCESS:

- **Excessive testing**: While testing is an important part of app development, excessive testing can delay the release of the app and lead to increased costs.
- **Overplanning and excessive documentation** can obstruct the development process, causing delays, wasting time and resources, and preventing the team from making headway.
- **Micromanagement**: Micromanaging the development team can hinder creativity and productivity and can lead to frustration and burnout among team members.
- **Over-engineering**: Over-engineering or building complex features that are not essential to the app's core functionality can increase development time and cost without providing significant value to users.
- **Insufficient communication among team members** can result in misunderstandings, delays, and mistakes in the development process. Therefore, it's crucial to establish effective communication channels and conduct regular check-ins to ensure that everyone is in sync.
- **Obsession with details**: Excessive focus on achieving perfection and getting caught up in minor details can hinder the development process, leading to delays and impacting the timely delivery of a functional product by the team.
- **Inflexible project management**: Being too rigid in project management can prevent the team from adapting to changes in requirements or unexpected challenges. It's important to have a flexible project management approach that allows for adjustments to be made as needed to keep the project on track and ensure a successful outcome.

Based on the necessary needs from the organizational process, we have identified the importance of maintaining project timelines, regular communication with stakeholders, and continuous feedback loops throughout the development process. To achieve these goals, we have decided to use the Scrum agile methodology as it provides a framework for iterative development and emphasizes collaboration and flexibility.

5.2) Agile Process Model (framework: Scrum)

5.2.1) SOFTWARE PROCESS DESCRIPTION:

Agile Scrum is a way to build software that emphasizes teamwork, communication, and being adaptable. Instead of planning everything out at the beginning, Agile Scrum breaks the work into smaller parts called sprints, which are usually one to four weeks long. During each sprint, the team works on finishing a small piece of the software.

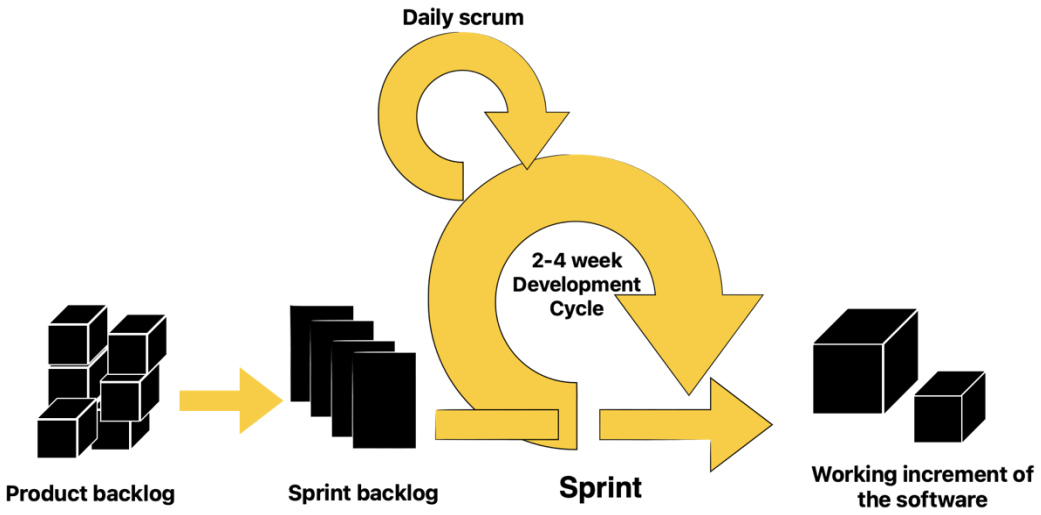
The team has people with different skills, like programmers and designers, who work together to finish each sprint. The person who decides what needs to be done is called the product owner, and they make sure the team is building things that are important. The scrum master is like a coach who helps the team stay organized and fix problems. Each day during a sprint, the team has a short meeting to talk about what they did the day before, what they plan to do that day, and if they need help with anything. This helps the team stay on track and know

what everyone else is doing.

At the end of each sprint, the team has something new that works, and they can show it to other people. This helps make sure the team is building the right thing and that the software is getting better each time.

Agile Scrum is good because it helps teams work better together and be more flexible. By breaking the work into smaller parts, the team can get things done faster and make changes more easily.

-In the following graph, you can find a high-level overview of the Agile process model and its various stages:



Graph 1: Agile process Model (Diagram is made by our team using "Freeform" application.)

5.2.2) How Agile Model Offers More Advantages for Software Development Projects

1. Faster Time-to-Market: Agile methodology allows for faster development cycles, which means that the mall navigation app can be delivered to the market more quickly. This is particularly important in today's fast-paced and competitive business environment.
2. Increased Transparency: Scrum framework promotes transparency by providing regular updates on the progress of the project. This enables the development team to identify and resolve any issues or bottlenecks in a timely manner.
3. Flexibility to Change: Agile methodology allows for changes to be made throughout the development process. This means that if the requirements or priorities of the mall navigation app change, the development team can easily adjust the project scope and deliverables.

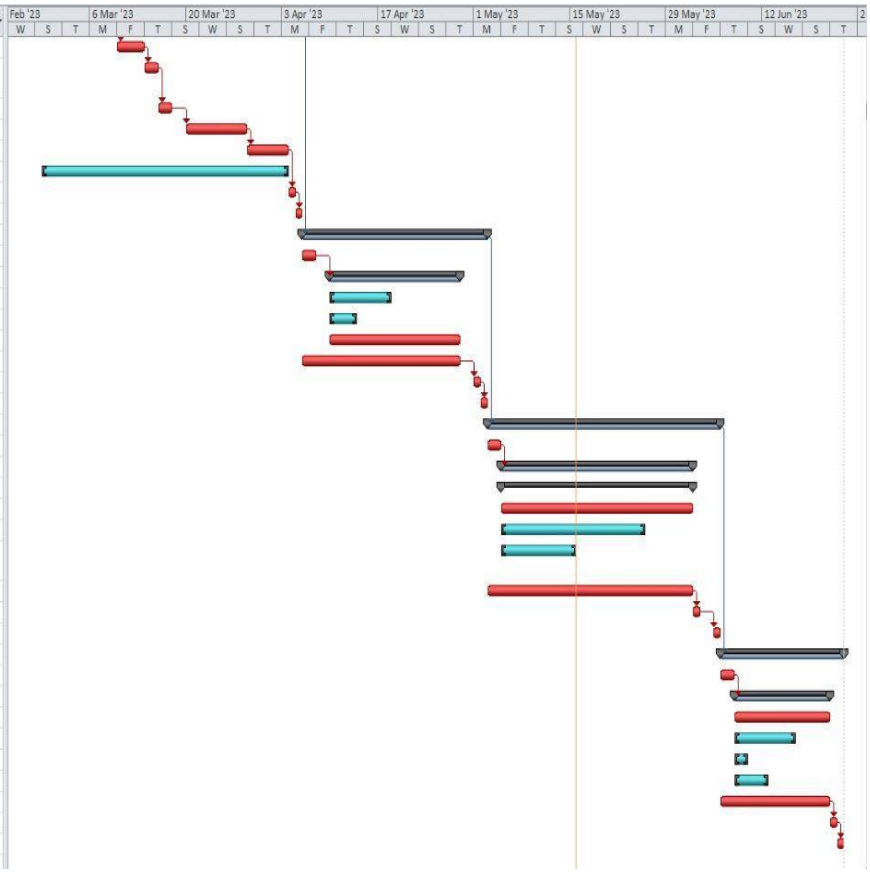
4. Higher Quality: With Agile methodology, testing is an ongoing process throughout the development lifecycle. This helps to ensure that the mall navigation app is of high quality and meets the needs of its users.

Overall, Agile process model with Scrum framework provides a collaborative, flexible, and iterative approach to software development, which is well-suited for developing a mall navigation app.

6. Schedule

The schedule section of development plan outlines the timeline and tasks associated with the implementation of our project using the **Scrum Agile** methodology. The schedule is organized into four sprints, each representing a focused iteration of development. In the following **Gantt chart**, you will find a visual representation of the timeline, with the associated tasks and activities clearly defined for each sprint. This schedule ensures that our team follows an iterative and incremental approach, allowing for regular feedback and adaptation to deliver high-quality software within the planned time frame.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
5		Elicit Requirements	2 days	Fri 10.03.23	Mon 13.03.23	4
6		Arrange Meetings With Stakeholders	2 days	Tue 14.03.23	Wed 15.03.23	5
7		Have meetings	2 days	Thu 16.03.23	Fri 17.03.23	6
8		Finalize Requirements	7 days	Mon 20.03.23	Tue 28.03.23	7
9		Prepare Project Documents	4 days	Wed 29.03.23	Mon 3.04.23	8
10		Daily Scrums	26 days	Mon 27.02.23	Mon 3.04.23	
11		Sprint Review	1 day	Tue 4.04.23	Tue 4.04.23	9
12		Sprint Retrospective	1 day	Wed 5.04.23	Wed 5.04.23	11
13		**Sprint2**	**19 days**	**Thu 6.04.23**	**Tue 2.05.23**	**1**
14		Sprint Planning	2 days	Thu 6.04.23	Fri 7.04.23	
15		**Design**	**15 days**	**Mon 10.04.23**	**Fri 28.04.23**	**14**
16		Visual Design	7 days	Mon 10.04.23	Tue 18.04.23	
17		Logo Design	4 days	Mon 10.04.23	Thu 13.04.23	
18		Interface Drawings	15 days	Mon 10.04.23	Fri 28.04.23	
19		Daily Scrums	17 days	Thu 6.04.23	Fri 28.04.23	
20		Sprint Review	1 day	Mon 1.05.23	Mon 1.05.23	19
21		Sprint Retrospective	1 day	Tue 2.05.23	Tue 2.05.23	20
22		**Sprint3**	**24 days**	**Wed 3.05.23**	**Mon 5.06.23**	**13**
23		Sprint Planning	2 days	Wed 3.05.23	Thu 4.05.23	
24		**Implementation**	**20 days**	**Fri 5.05.23**	**Thu 1.06.23**	**23**
25		**Mobile Programming**	**20 days**	**Fri 5.05.23**	**Thu 1.06.23**	
26		Java Programming Sessions	20 days	Fri 5.05.23	Thu 1.06.23	
27		Database Programming Sessions	15 days	Fri 5.05.23	Thu 25.05.23	
28		Integrating Navigation to Application	7 days	Fri 5.05.23	Mon 15.05.23	
29		Daily Scrums	22 days	Wed 3.05.23	Thu 1.06.23	
30		Sprint Review	1 day	Fri 2.06.23	Fri 2.06.23	29
31		Sprint Retrospective	1 day	Mon 5.06.23	Mon 5.06.23	30
32		**Sprint4**	**14 days**	**Tue 6.06.23**	**Fri 23.06.23**	**22**
33		Sprint Planning	2 days	Tue 6.06.23	Wed 7.06.23	
34		**Testing**	**10 days**	**Thu 8.06.23**	**Wed 21.06.23**	**33**
35		User Tests	10 days	Thu 8.06.23	Wed 21.06.23	
36		Admin Tests	7 days	Thu 8.06.23	Fri 16.06.23	
37		Translation Controls	2 days	Thu 8.06.23	Fri 9.06.23	
38		Navigation Data Tests	3 days	Thu 8.06.23	Mon 12.06.23	
39		Daily Scrums	12 days	Tue 6.06.23	Wed 21.06.23	
40		Sprint Review	1 day	Thu 22.06.23	Thu 22.06.23	39
41		Sprint Retrospective	1 day	Fri 23.06.23	Fri 23.06.23	40



progress, quality, and team performance. By collecting this data, we will be able to identify areas that need improvement, adjust our approach, and ensure that we deliver a high-quality product that meets the needs of our stakeholders.

7.1) Questions to identify measurements and Identified measurements associated with them:

1-How many defects were discovered during testing and how quickly were they resolved?

Identified Measurements

-Total number of defects discovered during testing.

-Time taken to resolve each defect.

-Number of defects that required rework.

Measurement storage and collection:

o What – Defects discovered during testing, time taken to resolve each defect, number of defects that required rework

o When – Immediately following each testing phase.

o Format – Real numerical data

o How - Entered a pre-specified project spreadsheet by the testing team

2-Was the project completed within its established schedule?

Identified Measurements

-Actual project completion date

-Project duration

-Percentage of scheduled tasks completed on time.

Measurement storage and collection:

What – Project completion date, project duration, percentage of scheduled tasks completed on time

When – Upon project completion and at regular intervals throughout the project

Format – Real number data

How - Entered in pre-specified project spreadsheet by the project manager

3-What is the quality of the final product?

Identified Measurements

-Number of defects discovered in the final product.

-Customer satisfaction ratings for the final product

-Number of support calls or tickets raised for the final product.

Measurement storage and collection:

What – Defects discovered in the final product, customer satisfaction ratings, number of support calls or tickets raised

o When – After the final product is released to customers.

o Format – Real number data

o How - Entered to pre-specified project spreadsheet by the support team

4-What amount of time was dedicated to testing?

Identified Measurements

-Total number of hours spent on testing.

-Time spent on each testing phase.

-Time spent on manual testing versus automated testing.

***Measurement storage and collection**:*

What – Total number of hours spent on testing, time spent on each testing phase, time spent on manual testing vs automated testing

When – Throughout the testing phase

Format – Real number data

How - Entered a pre-specified project spreadsheet by the testing team

5-How much documentation was required for the project?

Identified Measurements

-Total number of pages of documentation produced.

-Time spent on creating documentation.

-Percentage of documentation completed on time.

***Measurement storage and collection**:*

What – Total number of pages of documentation produced, time spent on creating documentation, percentage of documentation completed on time

When – Throughout the project

Format – Real number data

How - Entered to pre-specified project spreadsheet by the documentation team.

6-How much of the code has been reused?

Identified Measurements

-Number of code modules reused.

-Number of lines of code reused.

-Percentage of reused code in the final product

Measurement storage and collection:

What – Percentage, number of modules, and number of lines of reused code

When – After the final product is completed.

Format – Real number data

How - Manually extracted from code documentation and stored in a database.

7-Was the project completed within the allocated budget?

Identified Measurements

-Actual cost of the project

-Estimated cost of the project

-Variance between actual and estimated costs

Measurement storage and collection:

What – Actual and estimated costs, and cost variance

When – At the end of the project

Format – Real number data

How - Entered into a pre-specified project spreadsheet by project manager

8-What was the team's productivity during the project?

Identified Measurements

-Number of completed tasks.

-Time taken to complete each task.

-Number of defects per unit of work

Measurement storage and collection

What – Number of completed tasks, time taken to complete each task, defect density

When – Throughout the project

Format – Real number data

How - Automatically collected by project management software.

9-How effective were the development methods and tools used in the project?

Identified Measurements

-Number of defects introduced by the development tools.

-Time taken to complete tasks using the development tools.

-Satisfaction survey results from team members

Measurement storage and collection

What – Number of defects, time taken, survey results o When – Throughout the project

Format – Real number data and survey results

How - Collected manually by the project manager

10-How well did the software product perform under different conditions?

Identified Measurements

-Memory usage under heavy load

-Response time under light load

-Response time under heavy load

Measurement storage and collection

What – Response time and memory usage

When – During performance testing

Format – Real number data

How - Automatically collected by performance testing software.

11-How well did the project team communicate with each other and with stakeholders?

Identified Measurements

-Number of communication breakdowns within the project team

-Response time to stakeholder queries or issues

-Level of stakeholder satisfaction with project communication

Measurement storage and collection

What – Communication breakdowns, response time, stakeholder satisfaction

When – Throughout the project and at project completion

Format – Categorical and ordinal data

How - Collected through surveys, interviews, and project logs and stored in a project database

12-How well were project risks identified and managed?

Identified Measurements

-Number and severity of risks identified during the project

-Percentage of identified risks with mitigation plans

-Number of high-severity risks that occurred during the project

Measurement storage and collection

o What – Identified risks, mitigation plans, high-severity risks

o When – Throughout the project

o Format – Real number data and categorical data

o How - Risks and mitigation plans are documented and stored in a project

database, and high-severity risks are also recorded separately.

13-How well were project deliverables reviewed and approved?

Identified Measurements

-Number of deliverables reviewed

-Time taken to review each deliverable.

-Number of rework cycles for each deliverable

Measurement storage and collection

What – Deliverables reviewed, time taken to review each deliverable, number of rework cycles

When – After each deliverable is completed.

Format – Real number data

How - Entered to pre-specified project spreadsheet by the review team

14-What is the size of the software product?

Identified Measurements

-Lines of code (LOC)

-Number of functions or methods

-Number of classes or modules

Measurement storage and collection

What – Lines of code, number of functions or methods, number of classes or modules

When – At the end of each development phase

Format – Integer data

How - Calculated by a code analysis tool and recorded in a pre-specified project.

spreadsheet by a development team member

8. Project Risks

This section outlines potential risks that could impact the project's success. It is crucial for us to identify and prioritize risks early in the project to minimize their impact. In the table 1, we have identified and ranked potential risks based on both their likelihood and impact.

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISK DESCRIPTION
1	4	**5**	**Technical issues**: Technical issues, such as software bugs or system failures, can cause delays, increased costs, or even project failure
2	6	**8**	**Resource availability**: Insufficient or unavailable resources, such as personnel or equipment, can lead to project delays or quality issues.
5	3	**8**	**Budget overruns**: Exceeding the project budget can lead to project failure, financial loss, and damage to the organization's reputation.
3	7	**10**	**Requirements change**: Changes in the requirements can lead to scope creep, delays, and increased project costs.
4	8	**12**	**Schedule delays**: Delays in any part of the project can cause the overall schedule to slip, leading to increased costs and potential project failure.
7	5	**12**	**Quality concerns**: If the final product does not meet expected quality standards, it can lead to customer dissatisfaction, loss of business, and damage to the organization's reputation.
11	2	**13**	**Risk of competition**: If a competitor releases a similar app before ours, it can lead to decreased market share, loss of business, and potential project failure.
12	1	**13**	**User adoption**: If users do not adopt the app or do not find it useful, it can lead to low usage rates, decreased customer satisfaction, and potential project failure.
8	9	**17**	**Security risks**: The app may be susceptible to security breaches, leading to data loss, financial loss, and damage to the organization's reputation.
6	12	**18**	**Communication challenges**: Poor communication among team members, stakeholders, or customers can lead to misunderstandings, conflicts, and project failure.
9	10	**19**	**Insufficient training**: If end-users are not adequately trained on how to use the app, it can lead to low adoption rates, decreased user satisfaction, and loss of business.
10	11	**21**	**Staff turnover**: Key members of the project team may leave the organization, leading to delays and knowledge loss.

Table 1

9. Software Tools

The successful execution of a software development project relies on a combination of effective development practices, intuitive design principles, and meticulous documentation. In this section of the software development plan, we will delve into three critical tools that contribute to the overall success and quality of our project: Development (IDE), Design (UX/UI), and Documentation tools.

9.1) Development tools (IDE)

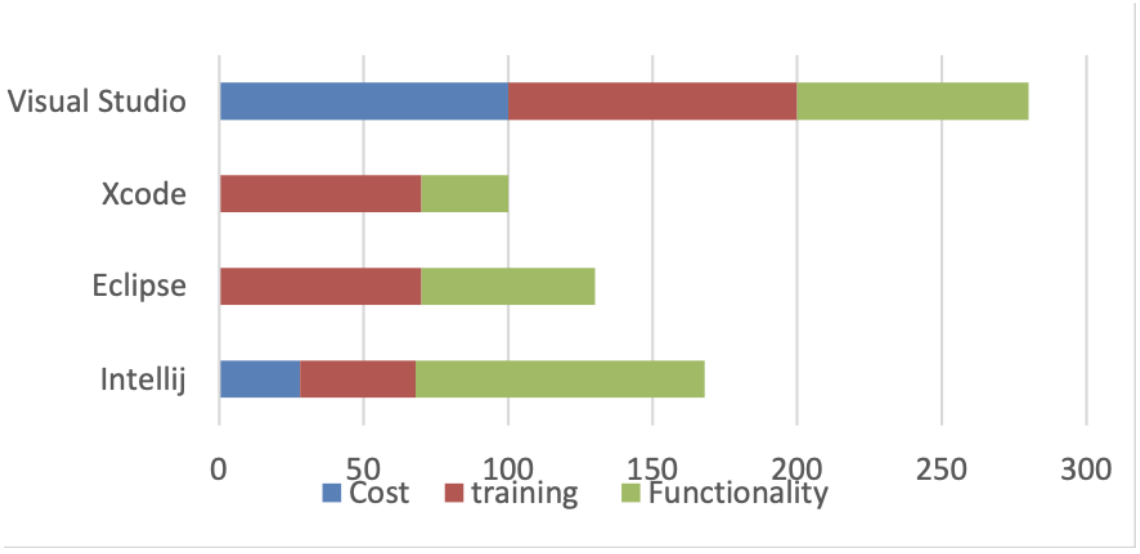
Tool Cost/Training/Functionality Data

Tool	IntelliJ IDEA	Eclipse	Xcode	Visual Studio
Cost	$70.00 per month	Open Source $0	Needs Mac, $0	$250 per month
Training Days	4	7	7	10
Functionality	100	60	30	80

Normalized Cost/Training/Functionality Data

Tool	IntelliJ IDEA	Eclipse	Xcode	Visual Studio
Cost	28	0	0	**100**
Training Days	40	70	70	**100**
Functionality	100	60	**30**	80

Normalized tool Graph



Our selected tool and reason for this choice

We have chosen IntelliJ IDEA as our development (IDE) tool for its distinct advantages over other options such as XCode, Eclipse, and Visual Studio. Firstly, IntelliJ IDEA provides cross-platform compatibility, allowing developers to work seamlessly across different operating systems. This flexibility ensures that our team can utilize their preferred operating system without any compatibility issues.

Additionally, IntelliJ IDEA offers a steep learning curve, providing a rich set of features that enhance productivity and efficiency. Compared to XCode, Eclipse, and Visual Studio, IntelliJ IDEA requires less training time for developers to become proficient. Its intuitive user interface and comprehensive documentation enable a quick grasp of the IDE's features, allowing developers to focus more on actual coding and project tasks.

9.2) Design tools

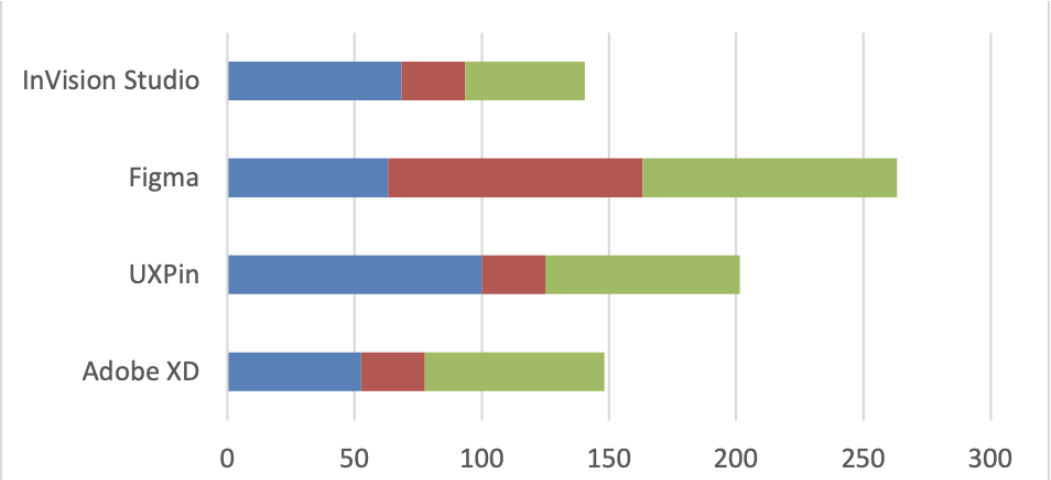
Tool Cost/Training/Functionality Data

Tool	Adobe XD	UXPin	Figma	InVision Studio
Cost	$9.99/month	$19/month	$12/month	$13/month
Training Days	1	1	4	1
Functionality	70	40	85	50

Normalized Cost/Training/Functionality Data

Tool	Adobe XD	UXPin	Figma	InVision Studio
Cost	52.6	**100**	63.2	68.4
Training Days	25	25	**100**	25
Functionality	70.6	76.5	**100**	47.1

Normalized tool Graph



Our selected tool and reason for this choice

We have selected Figma as our design (UX/UI) tool for several compelling reasons. Firstly, Figma offers free access for students, making it a cost-effective choice for our team. This allows students to leverage the full functionality of Figma without any financial constraints, promoting collaboration and learning among team members.

One of the key advantages of Figma is its accessibility. Being a browser-based tool, Figma can be accessed from any device with an internet connection, eliminating the need for installation or compatibility issues. This accessibility ensures that our design team can work seamlessly, regardless of their operating system or device, facilitating smooth collaboration and real-time feedback.

Furthermore, Figma's use of vector-based graphics is a significant advantage. This feature allows designs to be easily scaled and adapted to different screen sizes and resolutions. Whether it's a mobile device, tablet, or desktop, Figma's vector-based graphics ensure that our designs retain their quality and integrity across various platforms, enhancing the overall user experience.

9.3) Documentation Tools

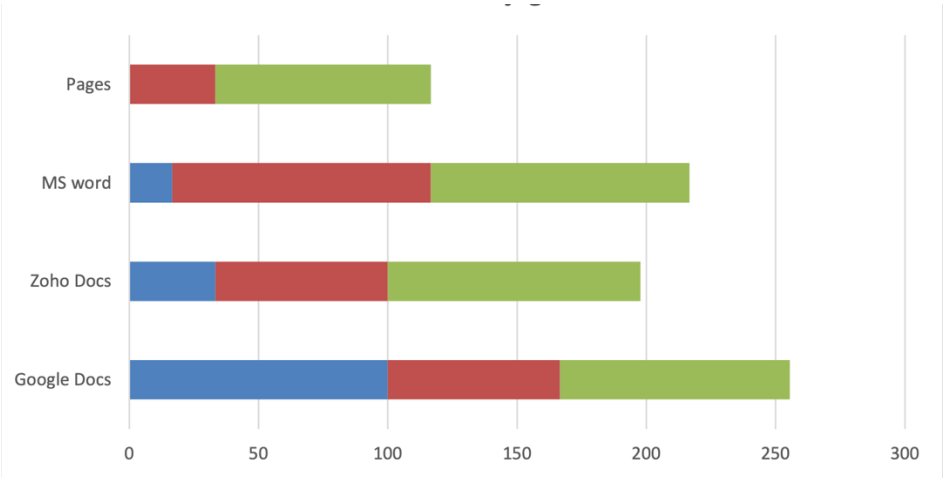
Tool Cost/Training/Functionality Data

Tool	Google Docs	Zoho Docs	MS word	Pages
Cost	$60/month	$20/month	$9.99/month	$0
Training Days	2	2	3	1
Functionality	80	88	90	75

Normalized Cost/Training/Functionality Data

Tool	Google Docs	Zoho Docs	MS word	Pages
Cost	**100**	33.33	16.65	Needs Mac, $0
Training Days	66.66	66.66	**100**	33.33
Functionality	88.88	97.77	**100**	83.33

Normalized tool Graph



Our selected tool and reason for this choice

We have chosen Microsoft Word as our documentation tool for several reasons. While it may require some additional time to become comfortable with its basic functionalities, Microsoft Word offers a range of advantages over its competitors such as Soho Docs, Google Docs, and Pages.

One of the key advantages of Microsoft Word is its affordable subscription fee. Compared to its competitors, Microsoft Word provides a cost-effective solution for our documentation needs. This allows us to leverage its extensive features and capabilities without incurring significant expenses.

Furthermore, Microsoft Word offers superior functionality compared to its competitors. It provides a comprehensive set of tools for formatting, editing, and organizing documents. Its robust features, such as advanced formatting options, collaboration tools, and integration with other Microsoft Office applications, enhance our ability to create professional documentation.

10. Project needs

While the tools we have chosen are instrumental in facilitating efficient development, intuitive design, and comprehensive documentation, it is imperative to consider the broader requirements essential for the successful execution of our project.

In this section, we will shift our focus towards comprehending the software needs that encompass the required functionalities, technologies, and platforms necessary to ensure the seamless operation of our software solution.

Furthermore, we will delve into the hardware needs that encompass the infrastructure, devices, and resources required to support the deployment and utilization of our software.

In addition to software and hardware, in this section we will also explore the support needs of our project.

10.1) Software Needs

IntelliJ IDEA

IntelliJ IDEA is an integrated development environment (IDE) that provides set of tools and features for developing software applications. It supports various programming languages, including Java. We will use IntelliJ IDEA as our primary IDE for developing the mall navigation app.

Android

Android is a mobile operating system developed by Google that is widely used on smartphones and tablets. As we plan to develop an Android version of the mall navigation app, we will need the Android Software Development Kit (SDK).

Figma

Figma is a design tool that we can use to create the visual layout and design of the mall navigation app. It allows us to collaborate with team members and create user-friendly designs that enhance the user experience.

Java SDK

Java SDK will provide us with the tools we need to develop and test our mall navigation app in the Java programming language. It includes tools for writing, testing, and debugging Java applications, as well as a Java Virtual Machine (JVM) that allows our app to run on different platforms.

Firebase

We will use Firebase as our cloud-based database solution for the app. Firebase's Realtime Database will allow us to store and retrieve data related to the mall, such as store information.

Git (Version Control System)

Git is the version control system we will use to manage and track changes to our app's source code. It will allow us to work collaboratively on the project, keep track of changes made by each team member, and revert to previous versions of code if needed.

Google Maps API

Google Maps API is a mapping and location-based service provided by Google that allows us to embed maps and location services into our application. With Google Maps API, we can integrate features such as search, directions, and location tracking into our applications.

10.2. Hardware Needs

Phones

A mobile phone is a crucial hardware requirement for developing the mobile app. It is essential to have a smartphone or multiple smartphones for testing and debugging the app on real devices. The phones have android operating system. we need phones with different screen sizes to ensure compatibility and a flawless user experience across different devices.

PC/Laptop

A PC or laptop is the primary hardware requirement for developing the mall navigation app. It should have sufficient processing power, RAM, and storage to run the necessary software tools and compile the app code.

All-in-One Printer

An all-in-one printer is a useful hardware requirement for developing the mall navigation app. It can be used for printing important documents related to the project, such as design specifications, user feedback reports, and progress updates. Additionally, it can be used for scanning physical documents and integrating them into the project management system.

PC Input Devices (Mouse, Keyboard, Microphone, etc.)

These are essential hardware tools that we will need for developing the application. Mouse and keyboard are essential for coding and testing the application. The microphone is necessary if we need to record audio or conduct virtual meetings with team members or stakeholders.

Tablet

A tablet can serve as a secondary device for testing and debugging the app on a portable platform, providing insights into the user experience on bigger screens. It can also be used for presenting and sharing design, as well as accessing project management tools and communication platforms on the go.

PC Output Devices (Monitors, Speakers, etc.)

Monitors provide us with a visual interface for working with the code and design application graphics. Speakers and headphones are necessary for testing the app's audio capabilities, such as sound effects, background music, or voice prompts.

10.3. Support needs

JetBrains Support Team (IntelliJ IDEA)

As we are using IntelliJ IDEA for app development, JetBrains Support Team can help us with any technical issues, bugs, or errors we encounter while using their software.

Microsoft Support Team (Office, Windows)

We are developing the mall navigation app on Microsoft's platforms, such as Windows and we also use Office for documentation, so MS support team can help us with our technical issues or questions related to using their products.

Investors

Investors can provide financial and strategic support for the mall navigation app development. They can help fund the project, offer guidance, and connect us with other resources and networks that can help us grow our business.

Firebase Support Team

Firebase's support team can help us with any technical issues or questions related to integrating Firebase services into our application.

Technical Writers

Technical writers can help us create documentation and user guides for the mall navigation app. They can write clear instructions and explain complex technical concepts in simple terms, making it easier for users to understand and use the app.

Marketing Experts

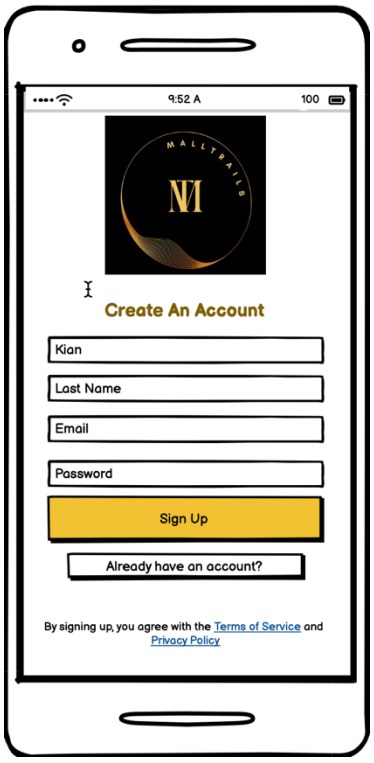
Marketing experts can help us create a marketing strategy for the mall navigation app and promote it to potential users. They can conduct market research and develop a social media and advertising campaign to increase app visibility and downloads.

User Experience (UX) Designers

UX designers can help us create a user-friendly interface for the app. They can conduct user research, create prototypes, and provide feedback and recommendations to optimize the app's user experience.

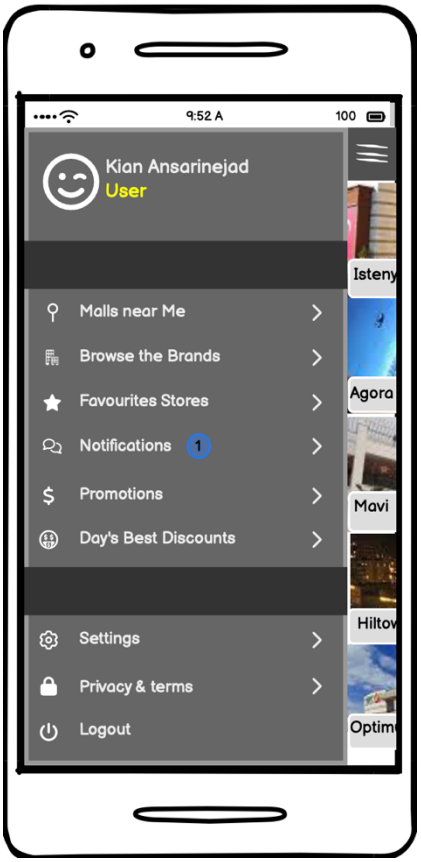
11. Graphical User Interfaces

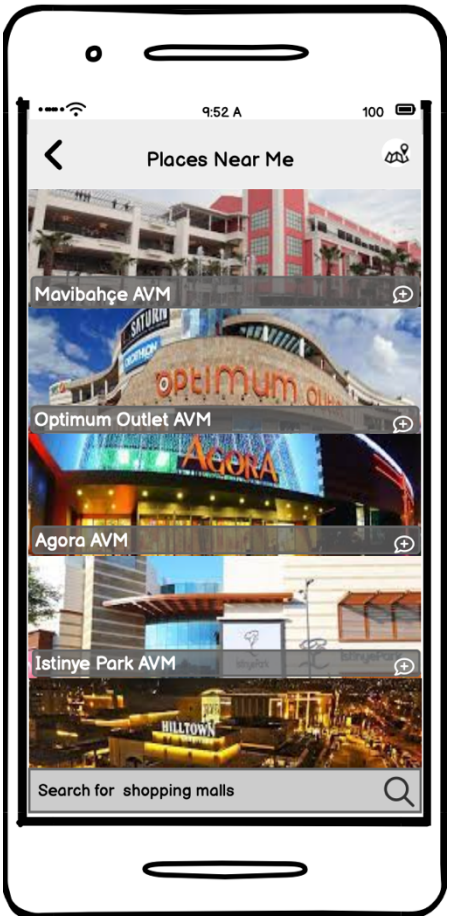
The graphical user interface (GUI) is a critical aspect of software development, serving as the visual and interactive medium for users to engage with our application. This section showcases sample pages of our application's GUI, highlighting key design principles, components, and functionalities.



The first page of our application's GUI features our application logo at the top, symbolizing our brand identity. Beneath the logo, users are greeted with a short form where they can enter their name, surname, email, and password to sign up as new users. Additionally, we provide a convenient option for returning users to sign in. This design ensures a streamlined and user-friendly experience, allowing both new and previous users to easily access and interact with our application.

This page displays the app's navigation bar that enhances user accessibility and engagement. At the top left, users can find their avatar and name, personalizing their experience within the app. Below that, a set of options are available, including "Malls Near Me," "Browse the Brands," "Favorite Stores," "Notifications," "Promotions," and "Day's Best Discounts." These options provide easy access to different features and functionalities of the application, allowing users to navigate effortlessly. Towards the end of the navigation bar, there are three additional options: "Settings," "Privacy," and "Logout." Selecting any of these options directs users to the corresponding page.

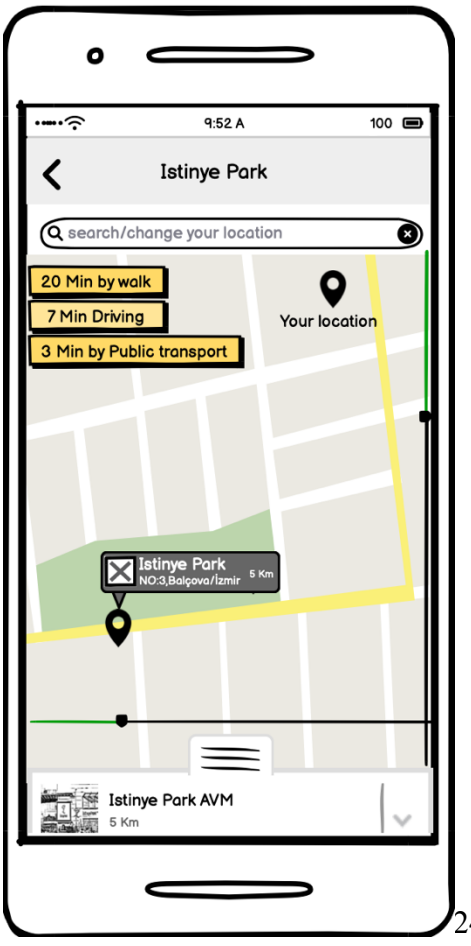




This page aims to assist users in discovering and exploring various shopping destinations based on their interests.

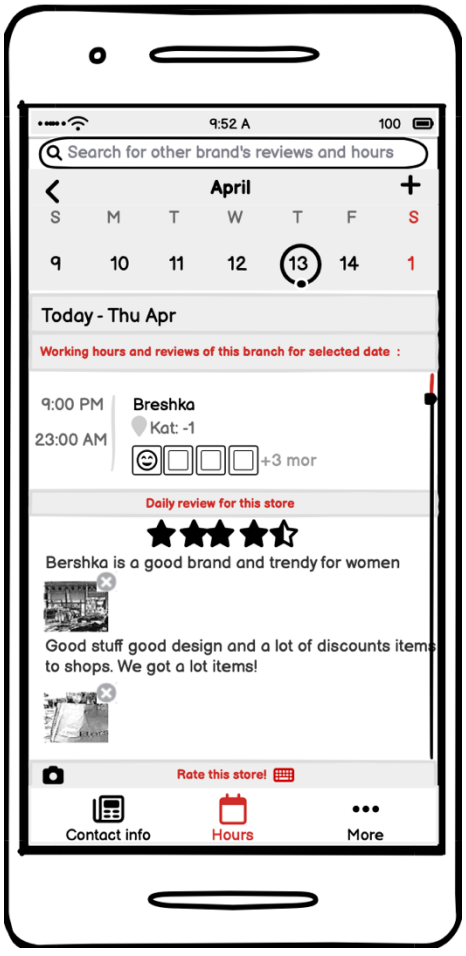
Users can view the names of the closest shopping malls along with a corresponding photo. Additionally, a convenient search bar is available, enabling users to easily search for specific shopping malls and access detailed information about them on the subsequent pages.

This page focuses on providing detailed information about a selected searched shopping mall. At the top of the page, users can find the name of the chosen mall, ensuring clear context for their search results. Below that, a convenient search bar allows users to modify their current location. By default, the app leverages the user's mobile GPS to determine their current geographical location, but users also have the option to manually input a different location. Once the location is set, users can view the shortest path to reach the shopping mall displayed on a map. The available transportation options such as walking, driving, or public transportation are provided, along with the estimated time each option takes. This page offers valuable information and navigation assistance, empowering users to efficiently plan their journey to the desired shopping mall.



The "Search for Brands" page of our application's GUI provides a convenient platform for users to find specific brands within the shopping mall. Users can enter the name of the brand they are looking for using the available search bar on this page. Below the search bar, a map is displayed, showcasing the directions from the user's current location within the mall to the searched brand. This feature ensures easy navigation and helps users locate their desired brand efficiently. Additionally, a list of other brands is presented below the map, allowing users to explore and access details about various brands within the mall. This page aims to simplify the brand search process and enhance the overall shopping experience.

The following page shoes calendar and a review section to provide users with information about brands within the shopping mall. The calendar initially displays the current date, but users can navigate to their intended days to check brand-specific details. Users can utilize the calendar to view the working hours and status of selected brands. For instance, it will indicate whether a specific store in the mall is open or closed on a given day or if there are any changes in working hours. Below this calendar section, users can access reviews for the selected brand within the chosen shopping mall. This feature allows users to read and write their own reviews, sharing their experiences with specific stores in the mall. Furthermore, users can find contact information for the stores, enhancing their ability to connect with the brands they are interested in.



Conclusion:

In conclusion, the mall navigation app project presented in this software development plan offers a comprehensive and all-in-one solution to the challenges faced by shoppers in large, complex shopping malls. Unlike existing apps in the market that cater to specific shopping malls, our app aims to provide a unified experience by integrating all shopping malls and their unique layouts.
Traditional methods of navigation, such as paper maps or asking for directions, often lead to frustration and wasted time for shoppers. However, with the mall navigation app, these inconveniences will become a thing of the past. Through its intuitive interface and turn-by-turn directions, users will effortlessly find their desired stores at any mall. The app's real-time location tracking ensures accurate guidance, even in the face of obstacles or closures, allowing users to optimize their shopping time.

Furthermore, the app goes beyond mere navigation, serving as a platform for discovering exciting deals and promotions within the mall. By actively promoting sales from participating stores, users will have the opportunity to explore new brands and products while enjoying significant savings. Using data analysis, the app will also provide personalized recommendations based on users' shopping history and preferences, making their experience more tailored and rewarding.

Throughout the project, various stakeholders will play crucial roles in its success. Shoppers/end users, project managers, development teams, mall management, retailers, marketing teams, data providers, investors, and competitors are all involved in different capacities, collectively contributing to the project's realization. This collaborative effort enables the app to address the diverse needs of the stakeholders.

In terms of project management, an agile process model, specifically the Scrum framework, has been adopted. This approach allows for continuous feedback and iteration, ensuring that the app evolves in response to user feedback and market demands. Additionally, the agile methodology facilitates easy maintenance and scalability, empowering the development team to deliver a robust and sustainable product.

Throughout the development journey, the project will rely on carefully selected software and hardware tools to optimize efficiency and productivity. IntelliJ IDEA, chosen as the development tool, enabling the development team to code with precision and speed. Figma, the design tool, offers a versatile platform for creating user-friendly interfaces, ensuring a seamless user experience. Microsoft Word has been chosen as the documentation tool, facilitating project documentation and effective collaboration among team members.

In measuring the project's success, various metrics will be assessed. These include the number of defects discovered during testing and the speed of their resolution, adherence to the established schedule, the quality of the final product, testing efforts, code reuse, and documentation requirements. These metrics will provide valuable insights into the project's performance, allowing for continuous improvement of the app.

In summary, the mall navigation app project promises to redefine the shopping experience in large malls. By providing efficient navigation, personalized recommendations, and essential information, the app aims to simplify and enhance the entire shopping journey for users. With agile development methodology, and carefully selected tools, the project is ready for success, benefiting shoppers, retailers, and mall management alike.