**Software Process Selection and Project Plan**

**09:PETS WALA**

| **Student ID** | **Name** |
| --- | --- |
| **22100004** | **Adil Aslam Chaudhry** |
| **22100111** | **Ayan Tabassum** |
| **22100201** | **Mohid Yousaf** |
| **22100298** | **Roshan A.Aziz** |
| **22100248** | **Sabahat** |

**Table of Contents**

[1. Introduction 3](#_gjdgxs)

[2. Software Process Selection 4](#_30j0zll)

[3. Gantt Chart](#_1fob9te) 7

[4. Risk Management](#_3znysh7) 7

[4.1 Potential Risks and Mitigation Strategies](#_2et92p0) 7

[5. Development Environment Preparation](#_tyjcwt) 9

[6. Deployment Platform](#_3dy6vkm) 11

[7. Who Did What?](#_1t3h5sf) 11

[8. Review checklist](#_4d34og8) 11

# Introduction

The project comprises an application catered towards pet owners, pet shops, sellers of pet accessories, veterinary doctors, pet rescue volunteers, and pet shelters/rescue teams. The application aims to link all these different entities through a variety of useful and helpful functionalities; pet owners would be able to search for different accessories, as well as finding good veterinary doctors for their pets, making their lives much more comfortable and better. They would also be able to interact with other pet owners and share photos and stories of their own pets as well. They could share helpful information with other pet owners. Any user of the app can also help with the animal rescue process by reporting pets to be rescued to the relevant rescue teams in the area.

# Software Process Selection

**2.1 Model Comparison**

**Waterfall Model:**

In a waterfall model, the project requirements, costs and timelines are worked upon in the start and are set in place. This is a more predictable approach where expectations and requirements are clearly communicated and are fixed in place. There is no room for change and the timelines are rigid. There is one final product that is delivered at the end of the whole process and it is subsequently tested.

**Pros:**

1. The client is given a clear expectation of when and what to expect for a finished product.
2. Rigid timelines allow for better planning and resource allocation
3. There is less dependency on specialized individuals and it is easy to find replacements
4. The system architecture can be specially designed since there is no need to account for changes.

**Cons:**

1. There is a lot of overhead time in starting the project as all requirements and design needs need to be figured out in the start.
2. Slow rollout of software since there is no iterative process and the product is only available at the end.
3. Cannot completely test products during production since there is no Working product. This it is not suitable for user centred design
4. There is no flexibility for change in client requirements. If the circumstances around the product change then there is no way to make changes to the requirements or design etc. The product direction cannot be realigned with respect to market changes.

**Agile:**

In the agile process, our project is divided into a large number of small tasks that are called sprints and for each sprint, a team works through a full software development cycle including planning, requirements analysis, design, coding, testing, accepting testing and release.

Each sprint is of a fixed time period i.e. four weeks and size of sprint is mostly based on team size (5-10 people).

**Pros:**

1. If the market for the software is poorly understood or the requirements are poorly understood, or expected to change, select a process that keeps flexibility, and we will use an iterative process containing sprints and phased implementation.
2. Revisits and rewrites of steps are encouraged to achieve the desired results.
3. Allowing for faster delivery and a better project.
4. Frequent delivery allows for quick changes in project direction while maintaining project scope.

**Cons:**

1. If a big software system has many interrelated components, and we want to avoid major changes to the design of a system during development, we can’t use agile methodology.
2. Agile doesn’t set a strict schedule, which, if not managed, can be difficult under a tight deadline.
3. Changing project requirements may cause problems in other areas of the organization.
4. Agile requires a consistent team. A weak link in the Agile team or management could result in wasted time and money.

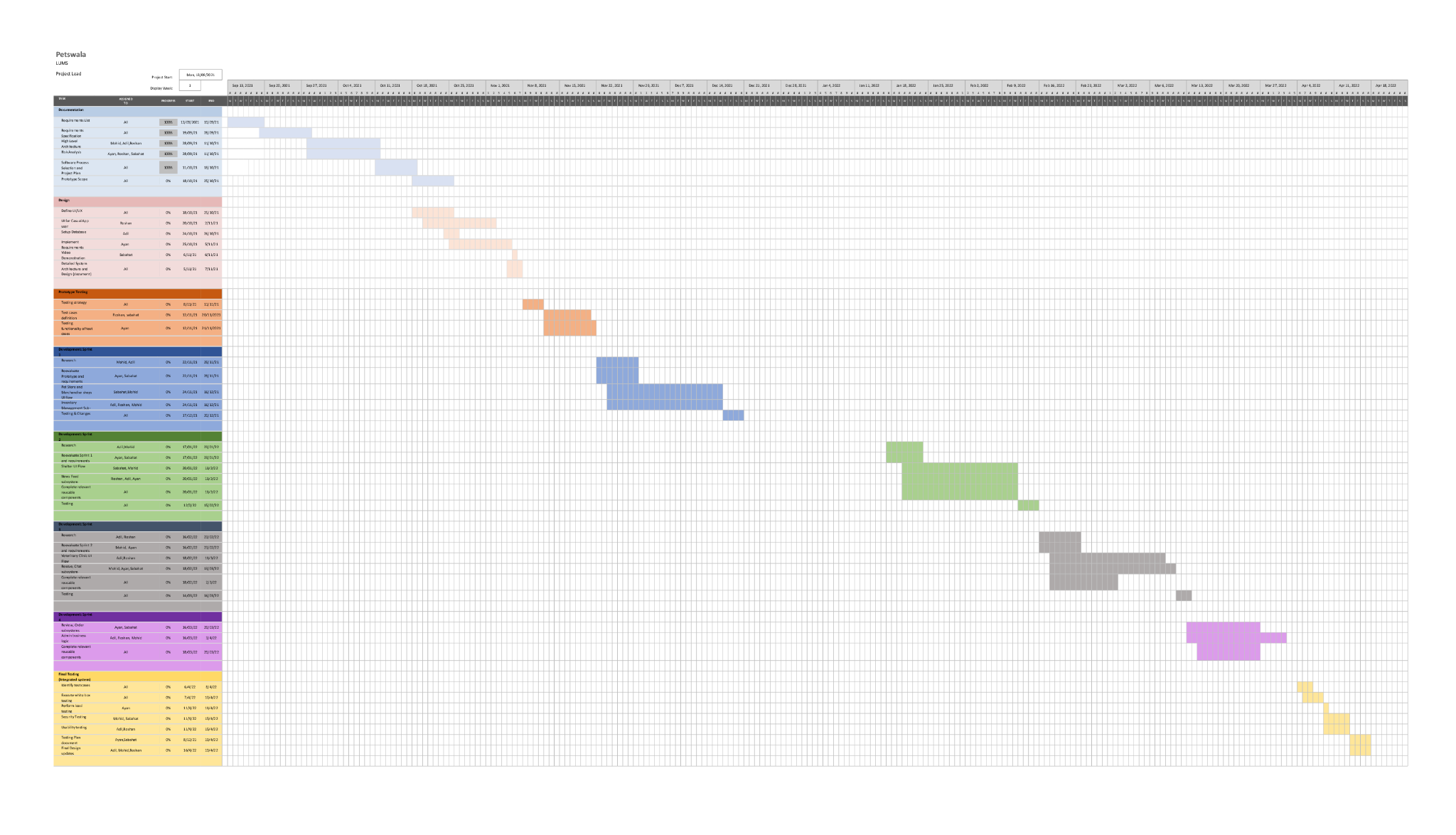
**2.2 Selection of process for project Development:**

We are going to use scrum methodology which is based on the agile philosophy.

**Justification:**

Our product is very modular in nature. We have a lot of subsystems which will be incrementally added with subsequent iterations. We also aim to incrementally make performance improving additions while focusing on delivering functionality first. We have a small fixed team which is perfect for an agile approach. We also aim to test our products after every sprint and incorporate a user-centred approach in order to maintain a correct direction with regards to fulfilling user needs. For this we need to be flexible in adapting to changing market dynamics. An agile approach is necessary to achieve this.

# Gantt Chart



# Risk Management

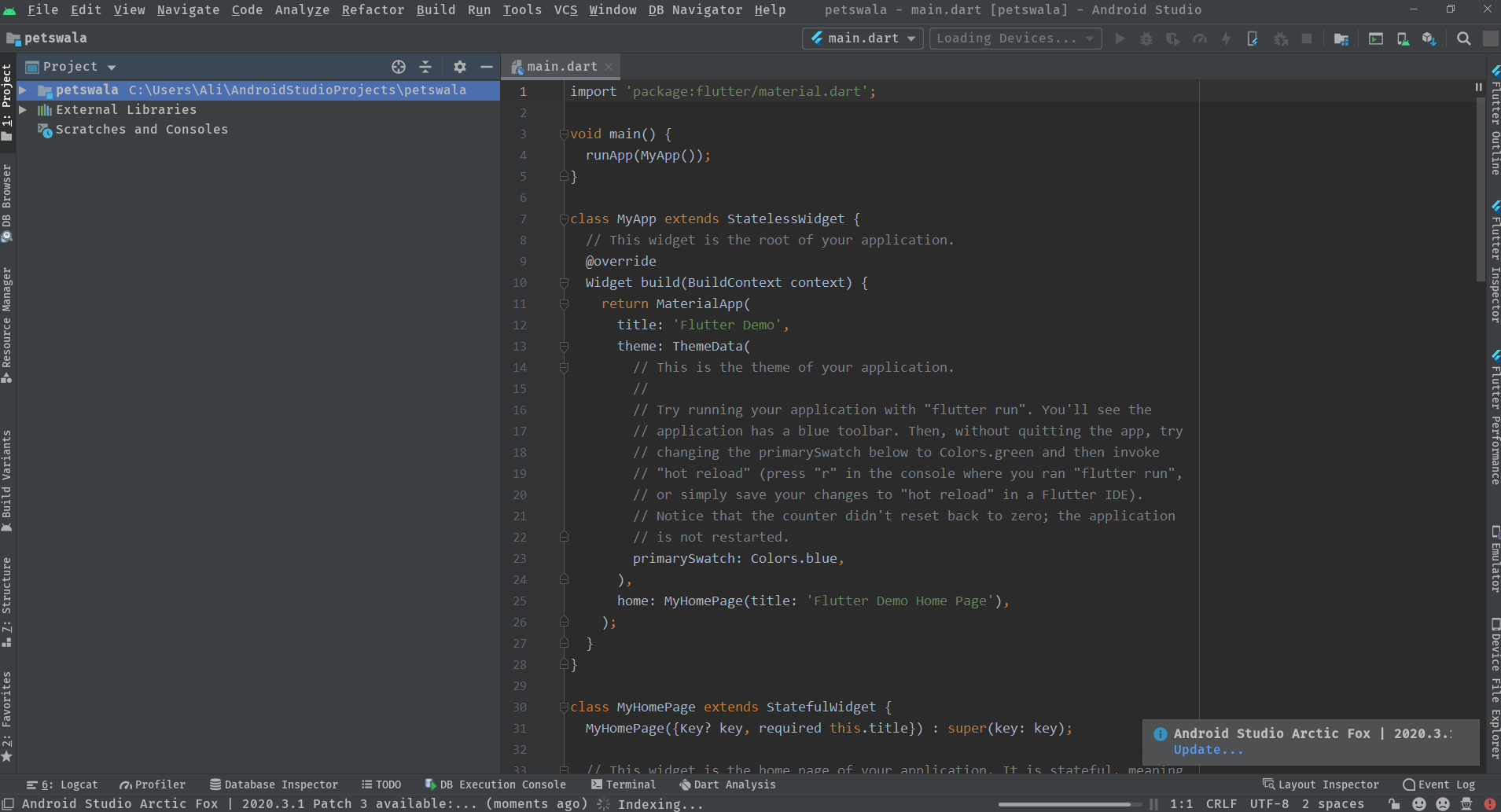
## 4.1 Potential Risks and Mitigation Strategies

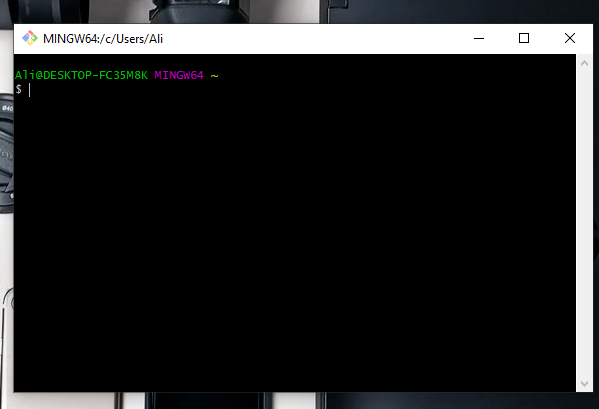
| **Sr.** | **Risk Description** | **Mitigation Strategy** |
| --- | --- | --- |
|  | User Engagement: After the release of the product, there might be a difference between stakeholders expectations and users actual engagement. | Incorporate user feedback in updates. Project will be designed in a way that is adaptable to changes. Moreover, testing, surveys and frequent updates will help avoid it. |
|  | Requirements Inflation: With the progress of the project, more requirements can be identified which were not stated in the beginning. | Involvement of customers at an early point in the development of the project, so that most of the requirements can be identified as soon as possible and relevant changes and estimations can be made. Moreover, frequent reporting in between the team would also help mitigate this risk. |
|  | Holes in software security: The application could have holes in it’s software security component and could be an unsecure application. | Use the appropriate APIs to have secure software. Additionally, plan your software architecture in a way that you have secure software. Logging the application use can also help in identifying these holes. |
| 4. | Unpredictable external risks: Sudden changes in the market, there can be a competitor with more resources and fast growth, some changes in consumer behaviour and priorities, or any new laws. | Thorough research about market, competitors and laws to mitigate the chances of such changes affecting the product. Enhance user experience and good marketing of the product |
| 5. | Lack of professional experience: Since we are students aiming to make a professional project, we are still learning. This inexperience could result in various problems. | Stay constantly updated on if we are on the right track by seeking help from seniors and the instructor and researching on Google. |
| 6. | Change in platform software: Due to the rapid changes in technology, changes can occur in the platform that we are developing for. Our application might risk falling off the cutting edge of technology and design. | Keep up to date on the release dates of these platform updates and either cater them during development or have an update path figured out beforehand. |
| 7. | Poor Management: Due to poor communication within the team, the developers might not share the important key information which is necessary for a smooth release of the product. | Scrum, weekly reports and frequent meetings |
| 8. | Resources: Unable to obtain required resources | Because the project does not require many resources, it is critical to obtain them as early in the development phase as feasible, and alternate resources, such as local test servers, should be prepared for. |
| 9. | Inaccurate Estimates: The estimates of number of users can be inaccurate. | We will do thorough research and conduct surveys to estimate the number of different types of users. |
| 10. | Code Quality: The lack of time or rushing, might result in poor code quality. | Leaving enough time to check for bugs, and do thorough testing |

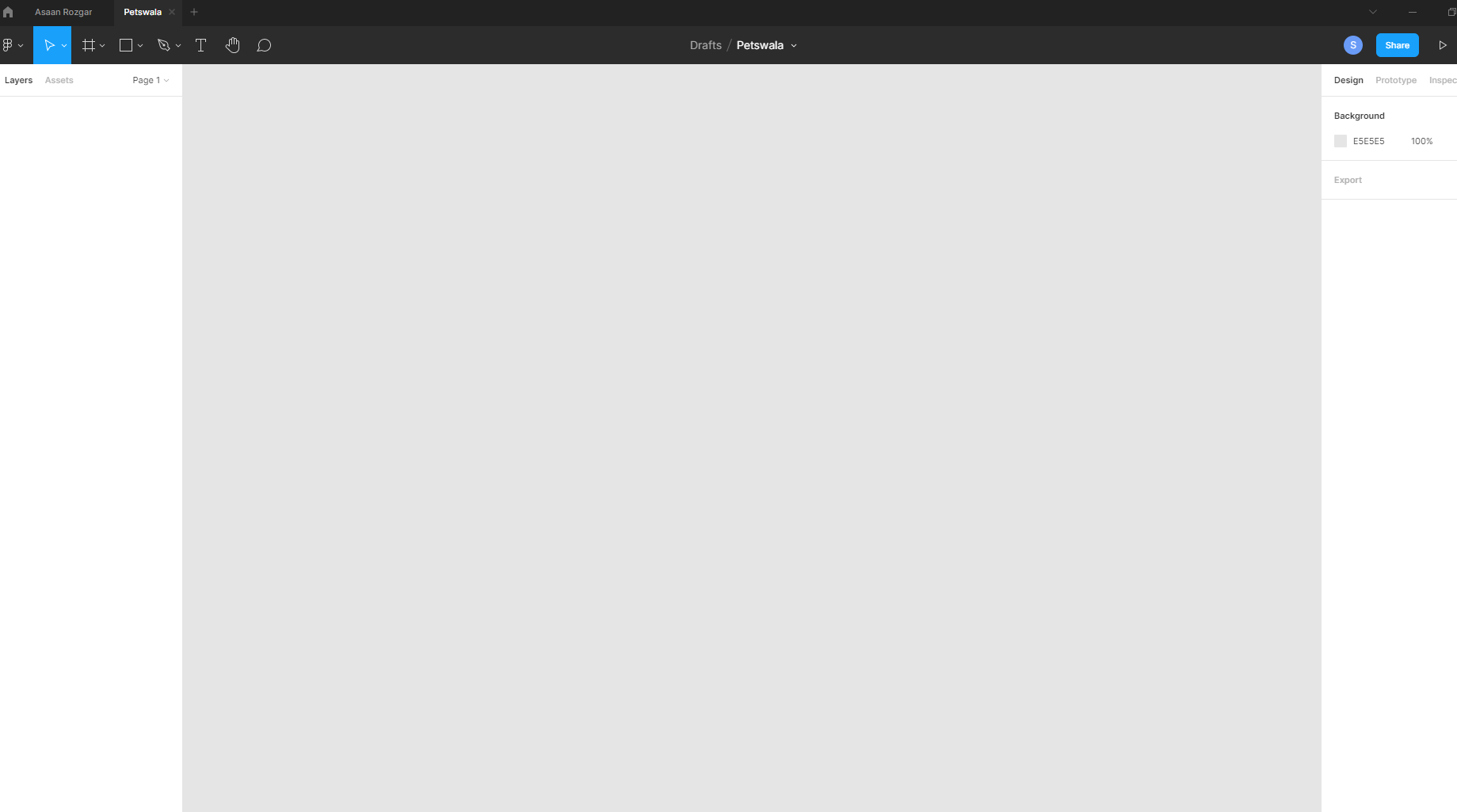
# 5. Development Environment Preparation

All mentioned development platforms have been prepared and are in use.

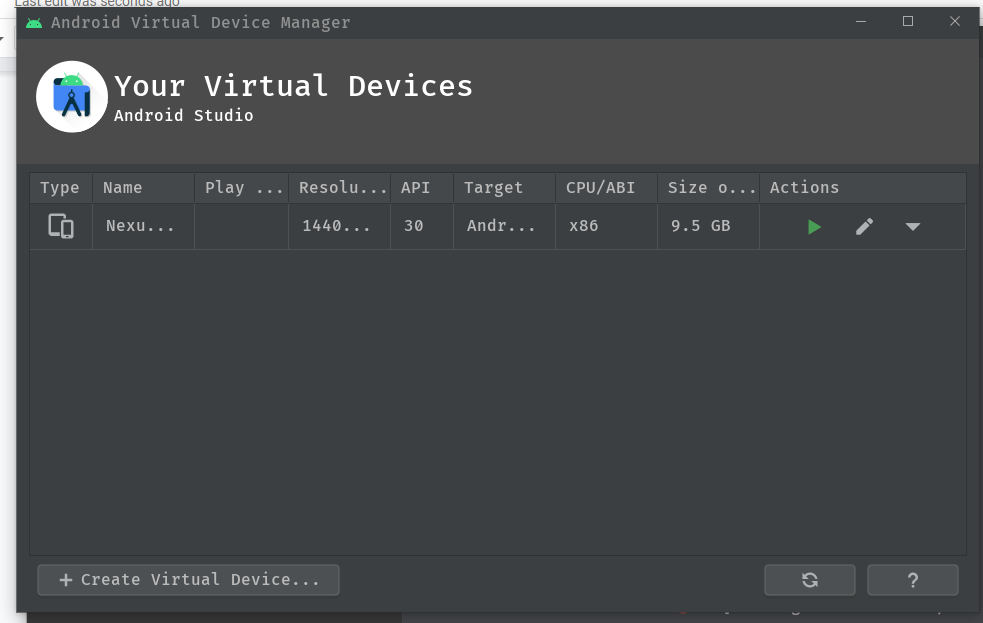
1. Flutter



1. Git Bash
2. Figma



1. AVD manager



# 

# 6. Deployment Platform

The APK for the app will be uploaded and made available for download online. The DB and backend will be temporarily hosted on a free hosting server with public access.

# 7. Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Adil Aslam | Software Process Selection, Deployment Platform, Discussion for Gantt Chart |
| Roshan Aziz | Discussion and development of gantt chart. |
| Mohid Yousuf | Software Process Selection, Deployment Platform, Discussion for Gantt Chart |
| Sabahat | Development of Gantt Chart, Discussion for Gantt Chart |
| Ayan Tabassum | Reviewed the Document and the Gantt Chart |

# 8. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Introduction | Sabahat |
| Software Process Selection | Adil, Mohid |
| Gantt Chart | Adil, Mohid, Ayan |
| Risk Management | Adil, Mohid, Ayan |
| Development Environment Platform | Adil, Mohid |
| Deployment Platform | Roshan |