**Batch: A1**

**Roll No.: 1911001, 1911003, 1911004**

**Chapter 1 Group No: 1**

|  |
| --- |
| **Title: Prepare documents related to your mini project** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Objective:** Prepare roadmap for developing a Mini project **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
|  | **At the end of successful completion of the course the student will be able to** |
| CO1 | Define the problem statement and scope of problem |
| CO2 | Identify various hardware and software requirements for problem solution |
| CO5 | Prepare a technical report based on the Mini project. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

**1. Life of SE Project**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction**:

**As studied in Software Engineering, developing a successful product (software: including the code and documents) needs a systematic approach.** In this experiment you will prepare the basic documents required to develop a product, a software system, a website or a mobile app to provide certain services or facilities.

Students will be required to prepare a document specifying.

**Objective of the project:**

Most users have thousands of images stored in their smartphone's photo gallery. This makes it difficult for users to find an image which they have saved, since unlike text documents, there is no robust method to search for images. With the users shifting towards cloud and having virtually unlimited storage capacity, this problem will be compounding rapidly in the future. This application aims to solve this problem.

To solve this problem, we are building a text-based image retrieval application, which links to a user’s photo gallery and generates text captions for the images stored in it. The users can then enter the text describing the events taking place in the image they are searching for; this application finds the closest captions stored to the query text and displays the images corresponding to them.

**Scope of the project:**

This project will include linking the developed app to the photo gallery app of the user. In addition to this two Machine Learning models will be developed: A caption generation model, which generates caption corresponding to the stored images and an NLP string similarity measuring model for displaying images having similar captions to the search query. The captions will be indexed to the images and stored on the user’s phone. The caption generation model will be deployed on a server.

This project however does not include the ability to store images for each user.

**Requirements gathering:**

Requirements for the project include:

Ram: 8 GB DDr3

CPU: Intel Pentium 4 processor or later that's SSE3 capable

Windows 7, Windows 8, Windows 8.1, Windows 10 or later

Browser: Chrome Version 92.0.4515.159, Edge Version 92.0.902.84

HDD: 4 GB free space

Modules

Django module for python 3

Pickle module for python 3

Pytorch module for python 3

Flask for python 3

Passlib for python 3

Psycopg2 for python 3

Difflib for python 3

React Native

Expo tools reactnative

Cloud support / local server for deployment of model

SQL database

Android 4.0+, 3GB ram, quad core processor

IOS 10+, 1.5 GB ram, A10

Complete Access to image folder

**Functional Requirements**

1. Text Retriever should be implemented successfully such that noise and unwanted text is not included
2. The software should be able to work with pictures taken from all cameras, as different users could use different snapshot devices, it should even be able to identify text in it
3. The software designed should be easy to use and should be able to be learnt quickly by all users.
4. Make app light weight so that all types of user depending on age/device/use type can use it.
5. Have a caption bank for caption wanted to be saved.

**Non Functional Requirements**

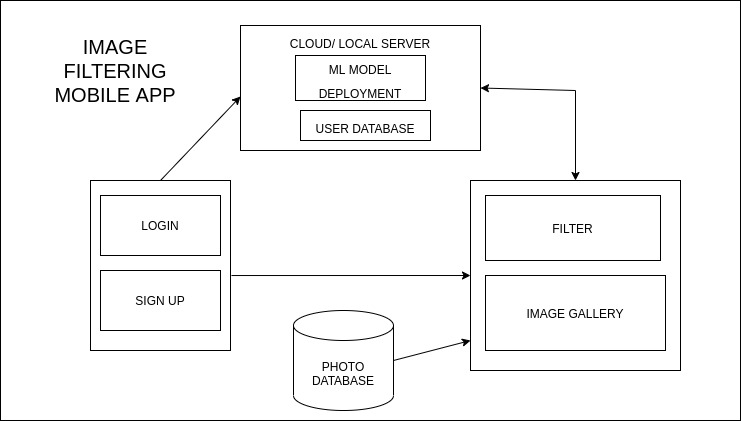
1. Aesthetic appearance should be implemented, something that is easy to use and the most important icons are at the forefront, it could be designed in a way that users like working and using it in convenient.
2. Fast image recognition system for quick and fast responses of the information.
3. Good data set to include as many possible images.

**Design:**

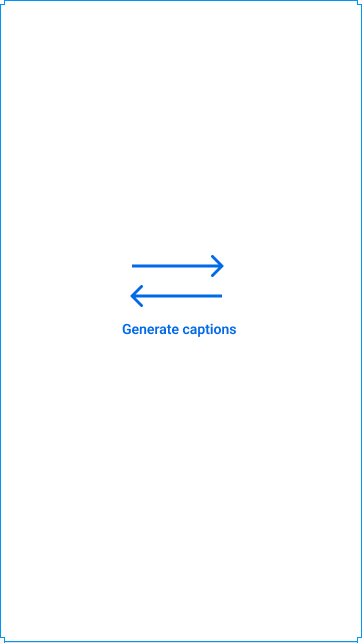
The product developed would be successfully developed if the requirements are analyzed properly and the design is proposed to implement properly. The modular approach / reusable/ pattern-based design might be followed for the various merits of each of them.

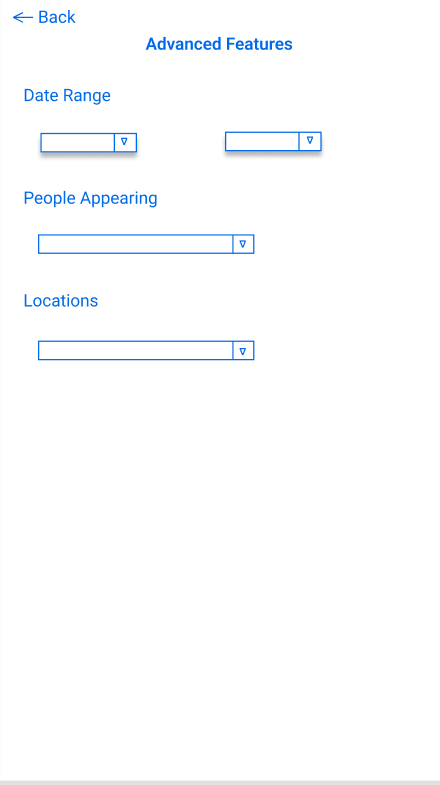
You will be required to provide Static & Dynamic Analysis Models to prepare the groundwork for development/ coding. This will add clarity as well as, if the development period is long and if the resources are changed, they can continue with the work.

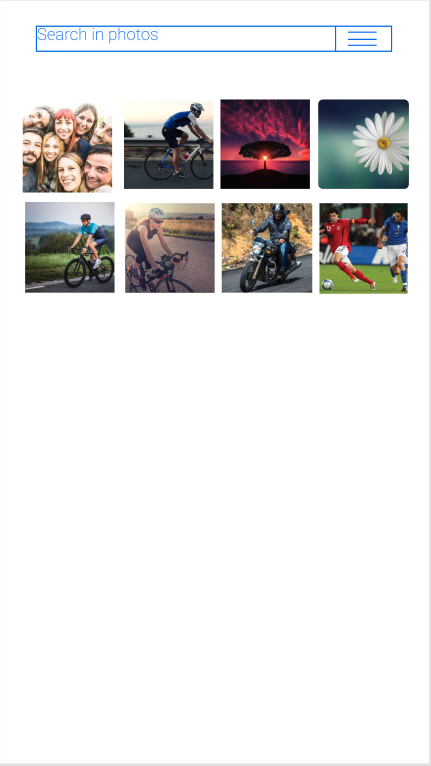
Outcome of this step could be a block diagram showing various parts (components/ modules) interaction with each other. Also Use case diagram (may be level 0), Interaction / Activity diagram, Class Diagram, Deployment diagram etc. whichever is applicable.

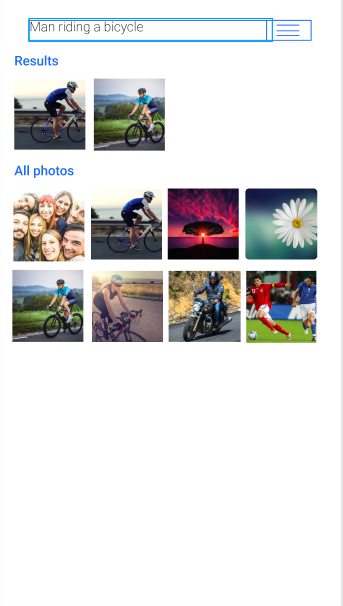
You may revise these diagrams as you progress.

UI design:



****



****

**Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | R1 | R2 | R3 | Mentor |
| **1.** **Requirement Gathering** |  |  |  |  |
| 1.1 Interaction with customer | C | C | C | A |
| 1.2 Preparing SRS | C | C | C | A |
| **2.** **Design** |  |  |  |  |
| 2.1 Preparing Block diagram | C | C | C | A |
| 2.2 Writing Functional Requirements | C | C | C | A |
| 2.3 Writing Non-Functional Requirements | C | C | C | A |
| 2.4 Developing Use Case | C | C | C | A |
| 2.5 Developing Test Cases | C | C | C | A |
| 1. **Planning** | C | C | C | A |
| 1. **Coding** |  |  |  |  |
| 4.1 NLP Model | C | R | R | A |
| 4.2 Front end/UI | R | C | C | A |
| 4.3 Integration | R | C | C | A |
| 1. **Testing** |  |  |  |  |
| 5.1 Unit 1 | E | A | A |  |
| 5.2 Unit 2 | A | E | E |  |
| 5.3 System Testing | E | E | E | A |

C: Creator, R: Reviewer, A: Approver E: Executor

TimeLine:

10th Jan - 10th feb 2022 : train model , prepare documentation for user, gather requirements

10th feb -28th feb : add various features and other image data for model and prepare it for accessing photos

28th feb-31st march : more documentation work , testing, integration with UI

31st march - 30th april : software testing (deployment) on various platforms

**Testing:**

The testing phase will include testing the application for accuracy, i.e. ensuring that the images similar to the search query are displayed; as well as ensuring that the app runs without any glitches. In addition to this we need to ensure that the server side of the app is able to successfully connect to each user and generate captions in real time.

**Implementation:**

**Requirement Document:** Based on the needs and functionality acquired, a requirement document should be created. This will be used to decide and estimate the duration of projects and human resources in the future. This data will be used to allocate tasks to various team members and maintain track of the tasks that have been performed.

**Implementation Plan:**

The app will be developed using React Native.

SQL will be used as the database.

For all the machine learning models, python will be used. All the models will be developed using the pytorch library.

**Step wise plan** :

1. train the model in tensor-flow object detection to choose the right text from image
2. make the model run until convergence and until the desired results and accuracy is not obtained.
3. create a copy of the model on your local pc and then train using your own images and images from other sources
4. integrate with the UI ad make sure it works properly

**Testing Plan:** Once the implementation is complete, new test cases will be built depending on the functionality of each module. A tester will be assigned to a team member who did not build the module and will be responsible for testing the module according to the test cases defined.

**Conclusion:**

We successfully prepared the SRS document for Image based text retriever project.

**Post Lab Activities (with reference to your tool):**

1. What are various Static & Dynamic Analysis Models?

Ans:

Various Static Analysis Models are:-

* + - * Control Analysis :-

This software focuses on examining the controls used in calling structure, control flow analysis and state transition analysis. The calling structure is related to the model by identifying the calling and call structure.

* + - * Data Analysis :-

Ensures proper operation is applied to data objects such as data structures and linked lists. In addition, this method also ensures that the defined data is used properly.

* + - * Fault/Failure Analysis :-

It analyses faults (incorrectly component) and failure (incorrect behaviour of model component) in the model. This method uses the input-output transformation description to identify the conditions that are cause for the failure.

* + - * Interface Analysis :-

This software verifies and verifies interactive and distribution simulations to check the code. There are two basic techniques for interface analysis and user interface analysis examines sub model interfaces and determines the accuracy of interface structure.

Various Dynamic Model are:-

* + - * Events:

An event is something that happen at a particular point in particular time such as a person press button. Event conveys information from one object to another.

The events are of three types: Signal event, Change event, and Time event.

* + - * State :

A state is an abstraction of attribute values and links of an object. Values and links are combined together into a state according to their entire behavior. The response of object according to input event is called state.

1. What are merits and limitations of Component based/ Pattern Based design approaches?

Merits:-

* + - * Ease of deployment − As new compatible versions become available, it is easier to replace existing versions with no impact on the other components or the system as a whole.
      * Reliability − The overall system reliability increases since the reliability of each individual component enhances the reliability of the whole system via reuse.
      * System maintenance and evolution − Easy to change and update the implementation without affecting the rest of the system.
      * Reduced cost − The use of third-party components allows you to spread the cost of development and maintenance.
      * Ease of development − Components implement well-known interfaces to provide defined functionality, allowing development without impacting other parts of the system.
      * Reusable − The use of reusable components means that they can be used to spread the development and maintenance cost across several applications or systems.
      * Modification of technical complexity − A component modifies the complexity through the use of a component container and its services.

Limitations:-

- The risk and challenges faced by developers that included freelance developers, IS departments, in developing components are also discussed.

- An application assembler locates suitable components and assembles them in integrated application systems that satisfy customer requirements. The customer employs component-based application systems to perform business tasks.

1. What are the attributes of good software?

* Functionality: It must be able to do what it was designed to do. The software requirements must guide the design and implementation of the software.
* Usability: The software must be usable; the users must not find it difficult to figure out how it works. It is user-cantered and user-friendly.
* Efficiency: Efficiency means that perform it's operations with minimal time and processing power. It uses the least amount of processing power and memory needed to achieve the desired result.
* Security: It must be secure. It should not cause physical or economic damage in the event of a system failure. Unauthorized users must not be allowed access to the system.
* Maintainability: It must evolve with changing requirements.
* Reliability: A reliable system will rarely fail, and even when it does fail, there are recovery mechanisms in the software to recover from the failure with minimal losses