

SPL-3 Technical Document

Anko (অঙ্ক)

A Handwritten Bangla Number Recognizer App

Course: SE 801

Submitted by

Maliha Nawshin Rahman (BSSE 0713)

BSSE Session: 2014-2015

Supervised by

Md Nurul Ahad Tawhid

Assistant Professor



Institute of Information Technology

University of Dhaka

[29-09-2018]

Letter of Transmittal

Md Nurul Ahad Tawhid

Assistant Professor

And

Dr. Naushin Nower

Assistant Professor

29 September, 2018

Dear Sir and Madam,

I have completed the Software Requirements and Design Analysis for my project Handwritten Bangla Number Recognizer android application 'Anko' and prepared the report on it having detailed information. This report includes Software Requirement Specifications and design plan for implementing the application. This report describes all the steps and sequential process I have followed while doing the analysis and design. Hence, I am submitting it to you for your approval.

Sincerely yours,

Maliha Nawshin Rahman (BSSE 0713)

Session: 2014-2015

Institute of Information Technology

University of Dhaka

Document of Authentication

This technical report of software project lab III on Handwritten Bangla Number Recognizer android application 'Anko' has been accepted and approved by the following-

Name	Title	Date	Signature
Maliha Nawshin Rahman	Analyzer and Developer	29.09.18	

Document Approval

Md. Nurul Ahad Tawhid

Project Supervisor

Signature

Institute of Information Technology

University of Dhaka

Acknowledgements

Undoubtedly it is a hard task to complete a project designing and analyzing the requirement for a software project perfectly and then preparing a report on it. I am happy that I have been able to complete this analysis as well as the report for the Handwritten Bangla Number Recognizer App by the grace of Almighty. I am also very thankful and grateful to my respected supervisor Md. Nurul Ahad Tawhid. It could be more difficult task for me if he would not share his valuable knowledge and lead to complete whole procedure wonderfully.

Abstract

In recent times, automatic handwritten character recognition is of great academic and commercial interests. The main challenge in handwritten character classification is to deal with the huge variety of handwriting styles by different writers in different languages. There are recent studies on Bangla handwritten digit recognition too but no app has been developed yet, to my best knowledge. This Android app will be developed as a tool for recognizing Bengali handwritten digit. It will let users take photo of any handwritten Bangla number and recognize it in different formats in English language. Bangla belongs to one of the most spoken language which ranks 5th in the world. About 220 million people use Bengali as written and spoken language in everyday life. (1) Therefore, automation of handwritten Bangla digit recognition(HBDR) has significant usage. This app will show the desired output of recent studies on Bangla hand written digit recognition easily on Android phone. It will also help foreigners who face difficulty in recognizing Bangla digits.

Contents

Chapter One	1
Introduction	1
1.1 Purpose	1
1.2 Intended Audience.....	1
Chapter Two	3
Inception	3
2.1 Introduction	3
2.2 Conclusion	4
Chapter Three.....	5
Elicitation	5
3.1 Introduction	5
3.2 Eliciting Requirements	5
3.3 Requirements Gathering.....	6
3.4 Quality Function Deployment.....	6
3.5 Usage Scenario.....	7
3.6 Elicitation Work Product	8
Chapter Four	9
Scenario Based Model	9
4.1 Introduction	9
4.2 Use Case Diagram	9
4.3 Activity Diagram.....	11
4.4 Conclusion	16
Chapter Five	17
Architectural Design.....	17

5.1 Introduction	17
5.2 Representing the System in Context	18
5.3 Defining Archetypes.....	19
5.4 Refining the Architecture into Components.....	20
5.5 Describing Instantiations of the System	21
5.6 Architectural Mapping Using Data Flow	22
5.7 Conclusion	23
Chapter Six	24
Conclusion.....	24
Appendix.....	25
References	25

Figures

Figure 1: Level 0 Use Case Diagram	10
Figure 2: Level 1 Use Case Diagram	11
Figure 3: Activity Diagram of Capture Photo Subsystem.....	12
Figure 4: Activity Diagram of Choose Photo Subsystem.....	13
Figure 5: Activity Diagram of Draw Digit Subsystem	14
Figure 6: Activity Diagram of View Subsystem	15
Figure 7: Steps of Architectural Design.....	17
Figure 8: Architectural Context Diagram	18
Figure 9: Archetypes	19
Figure 10: Overall Architectural Structure with Top-Level Components	20
Figure 11: An Instantiation of the Anko App system Architecture.....	21
Figure 12: Level 0 Data Flow Diagram	22
Figure 13: Level 1 Data Flow Diagram	22
Figure 14: Data Flow Diagram of Recognition process	23

Chapter One

Introduction

This chapter describes the objectives of this technical report as the audiences who should have to go through this report for individual purposes.

1.1 Purpose

This technical document is based on the Software Requirement Specification (SRS) and Software Design for the project- Handwritten Bangla Number Recognizer App. It includes all necessary requirements to develop this application no matter whether they are functional or non-functional. The information about the requirements here have been organized systematically so that everyone can easily figure out a summarized concept about the android app, Anko. This document serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer and stakeholder community. It will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This report is intended for several audiences, including the customer, as well as the project managers, developers, and testers.

- The customer will use this document to ensure that whatever she requires has been fulfilled by the project teams.
- The project managers of the developer team will use technical document to fix a milestone and time to deliver the software and to ensure that the teams working on this project are on the right path.
- The developers will use this report as a basis for developing the system's functionality. The developers will link the requirements defined in this document to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.

- The testers will use this document to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented here. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented here have been fulfilled.

Chapter Two

Inception

In this chapter, I will discuss about the first step of Software Requirements Specifications Analysis, that is, Inception.

2.1 Introduction

Requirement Engineering comprises several sequential steps. Inception is the first one among them. Inception creates the entrance to the project for the requirements analysts. It refers them how the project should get started. It also provides a basic idea to the engineers about the problems ahead which are needed to be solved and how critical obstacles may come during the project. (3) The main target of Inception phase is to identify the people related to the project and their needs. In order to complete this phase, I have focused on -

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the first questions

2.1.1 Identifying Stakeholders

Stakeholders are entities that have an interest in a given project. These stakeholders may be inside or outside an organization which:

- Sponsor a project, or
- Have an interest or a gain upon a successful completion of a project,
- May have a positive or negative influence in the project completion

There is one stakeholder for this system. That is an android app supplier who requested for the app to develop and design as well as specified the requirements for the app.

2.1.2 Requirements:

The requirements that have to be fulfilled in the application are:

- The app should be able to launch on any version of android smart phone.
- User can activate or deactivate the app from functioning.
- User can view the written or drawn Bangla number recognized in English
- The app should have a high accuracy in recognizing digit.

2.1.3 Asking First Questions

I set my first set of context-free questions focuses on the stakeholder, overall project goals and benefits. These questions helped to identify the measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped to gain a better understanding of problem and allows the customer to voice her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.2 Conclusion

Inception phase helped to establish basic understanding about the Handwritten Bangla Number Recognizer application, identify the people who will be benefited if the application is implemented, define the nature of the Anko app and establish a preliminary communication with the stakeholder.

Chapter Three

Elicitation

In this chapter, I will briefly discuss about the Elicitation phase of Anko application.

3.1 Introduction

Requirements elicitation is recognized as one of the most critical, knowledge-intensive activities of software development; poor execution of elicitation will almost guarantee that the final project is a complete failure. Since project failures are so rampant, it is quite likely that improving how the industry performs elicitation could have a dramatic effect on the success record of the industry. Improving requirements elicitation requires to first understand it. Although many papers have been written that define elicitation, or prescribe a specific technique to perform during elicitation, nobody has yet defined a unified model of the elicitation process that emphasizes the role of knowledge.

3.2 Eliciting Requirements

Earlier I have seen that the methodology used in Inception phase is Question and Answer approach. But Elicitation is quite different in this point of view. The elicitation phase follows a format of eliciting requirements which combines the other four phases namely problem solving, elaboration, negotiation and specification. In order to elicit requirements, I have followed four steps:

- Collaborative Requirements gathering
- Quality Function Deployment (QFD)
- Usage Scenarios
- Elicitation work product

3.3 Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. I completed following steps to do it:

- Meetings have been conducted with the android app supplier and she was questioned about her requirements and expectations from the Anko application.
- She was asked about the existing problems that are being faced by users without the application.
- Based on the meetings and response of her, I finally selected the requirements.

3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to my project, the following requirements are identified by a QFD.

3.4.1 Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of my project are:

- Capturing photo of handwritten Bangla digit
- Recognizing handwritten Bangla number (consists of more than 1 digit)
- Viewing the recognized handwritten Bangla digit in both English and Bangla

3.4.2 Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause for dissatisfaction. The expected requirements of my app are:

- Choosing photo of handwritten Bangla digit from photo gallery of the device
- Viewing the recognized handwritten Bangla digit in the English words format
- Drawing digit on the touch screen to take input from the android device

3.4.3 Exciting Requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. The exciting requirements for my app are following:

- Provide options to perform calculation (add, subtract, divide, multiply) with the drawn digits
- Update training model by adding new image data to the dataset

3.5 Usage Scenario

The aim of this android app is to recognize Bangla handwritten digit as well as number. Input of the number will be taken in three different ways- capturing image, choosing photo and drawing digit. Recognized digit will be provided in both word and number format in English language.

Capture Handwritten Digit

When user starts the app she will see few options for recognizing digit. Tapping the camera button icon will open android device's camera to capture photo. This option requires piece of paper as canvas for handwritten Bangla digit. The number might be consisted of one or more digits. While taking photo, user must focus on the handwritten digit of the paper and take a clear photo. This is because, blur photo may not give any result at all. Handwriting should also be accurate and good enough to understand.

Choose Numeric Photo

User can choose handwritten digit's photo from the device's photo gallery as well. The photo must contain Bangla digit written on paper. Image of any other object will not give expected result.

Draw Digit

Besides the photo options, a certain area (canvas) will be provided on the screen for the user. User can draw digit by moving finger on the touch screen. So, users will simply write the digit on canvas as they would on paper.

Clear Drawn Digit

If the drawn object does not match with any known Bangla digit, then it may not provide expected result. So if the drawn figure seems wrong, then user can clear the canvas by tapping a button. When the canvas is cleared, user can try to redraw the digit. After a successful recognition, user can draw again by clearing the canvas.

Tap to Detect

User must capture, choose or draw the digit to recognize. Then she needs to tap the detect button to confirm the digit or number for recognition. Then the confirmed digit will go through recognition process on the device.

View

It may take a little time to show the expected result. Result may vary on the quality of photo and drawing of digit. If it is not in recognizable shape, then a failure message will pop up and request the user to try again. Otherwise, recognized Bangla number will be shown in both English and Bangla digits. Besides the digit format, detected number will be expressed in English words as well.

3.6 Elicitation Work Product

The output of the elicitation task can vary depending on size of the system or product to be built.

My elicitation work product includes:

- Set of usage scenarios.
- Description of the system's technical environment.
- Make a bounded statement of scope for my system.
- Make a list of user and other stakeholder who participated in requirements.
- Make a statement of the requirements for Anko.

Chapter Four

Scenario Based Model

This chapter is about the scenario based model of the app, Anko:

4.1 Introduction

Scenario based modeling is an inexpensive rapid prototyping technique. This method is effective when systems are being built with the requirements vaguely known at the outset. Users are involved right from the start, to build prototypes evolving towards the final product. The users are also involved with the testing of the prototypes which is essential for the validation of requirements and help the users to gain an initial experience of the final system during the development itself. This method involves techniques which are applied by one or more professionals working alongside users who are expected to provide and specify their requirements at the beginning as well as evaluate and approve the system upon completion. (2)

4.2 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system.

The purposes of use case diagrams are:

- Gathering requirements of a system.
- Getting an outside view of a system.
- Identifying external and internal factors influencing the system.
- Showing the interaction among actors.

The first step in writing a use case is to define the set of actors that will be involved in the story. Actors are of two types. They are:

1. **Primary Actors:** Primary actors are the actors using the system to achieve a goal. They both consume data and produce information.
2. **Secondary Actors:** Secondary actors are the actors that the system needs assistance from to achieve the primary actor's goal. They either consume data or produce information.

Once actors have been identified, use cases can be developed.

4.2.1 Use Case Diagrams and Description of Subsystems

There are two actors in the system- user and system. The use case diagram of each subsystem and their description will be discussed in this section.

Level 0: Anko

Primary actor: User, System.

There are no secondary actors in Handwritten Bangla Digit Recognition System application.

Goal in context: The diagram in Figure 1 represents the whole Anko App System.

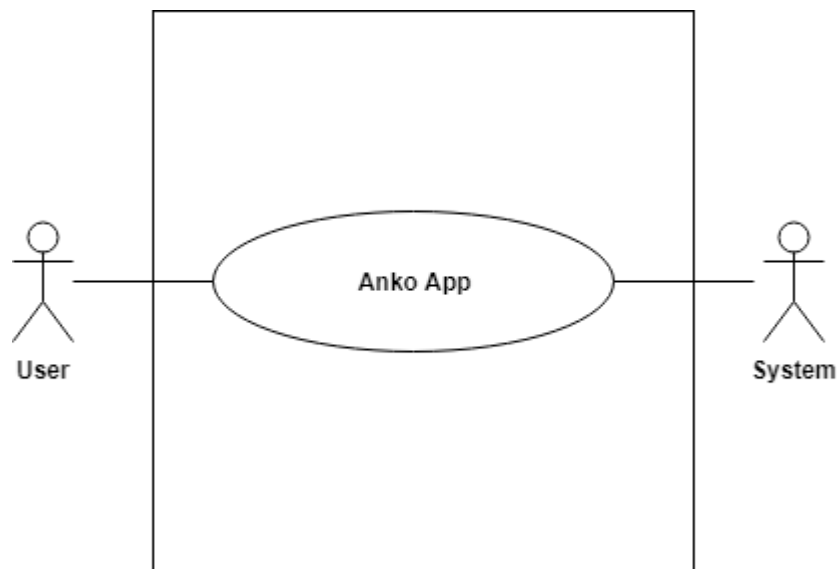


Figure 1: Level 0 Use Case Diagram

Level 1: Anko

There are four subsystems in Anko application. These are- Capture Photo, Choose Photo, Draw Digit and View. Figure 2 shows these subsystems.

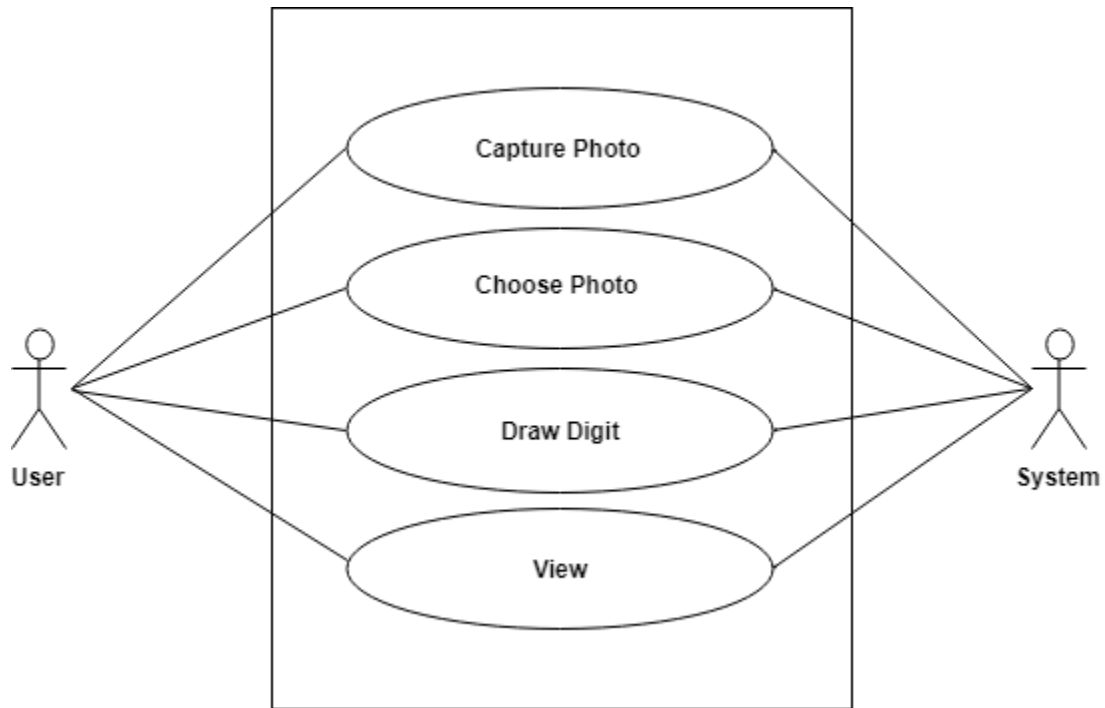


Figure 2: Level 1 Use Case Diagram

4.3 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join and so on.

(2)

4.3.1 Activity Diagrams of Subsystems

Subsystem: Capture Photo

Taking or capturing photo is the main option to take input of the digit to be recognized.

Following diagram on Figure 3 explains the activity of this subsystem.

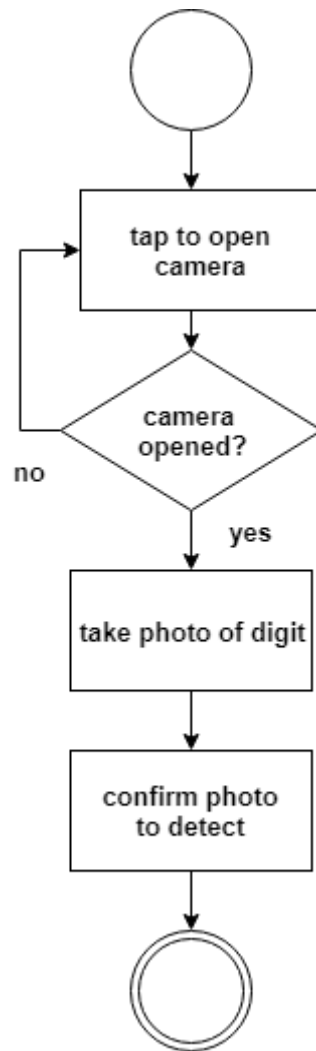


Figure 3: Activity Diagram of Capture Photo Subsystem

Subsystem: Choose Photo

Choosing photo is another option to take input of the digit to be recognized. Following diagram on Figure 4 explains the activity of this subsystem.

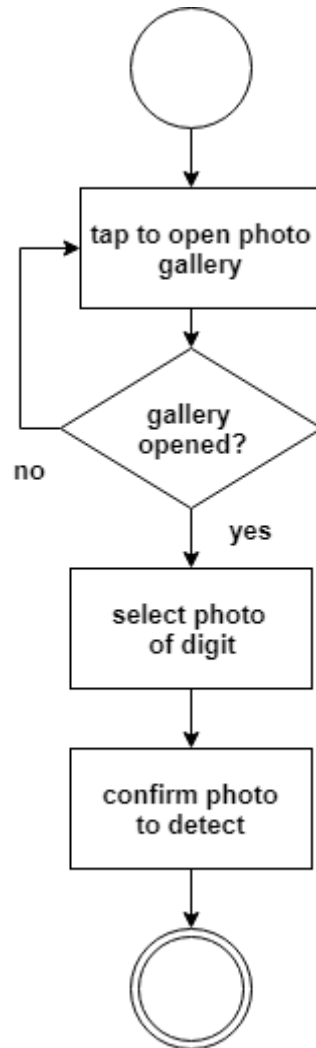


Figure 4: Activity Diagram of Choose Photo Subsystem

Subsystem: Draw Digit

Drawing digit on a canvas is an additional option to take input of the digit to be recognized.

Following diagram on Figure 5 explains the activity of this subsystem.

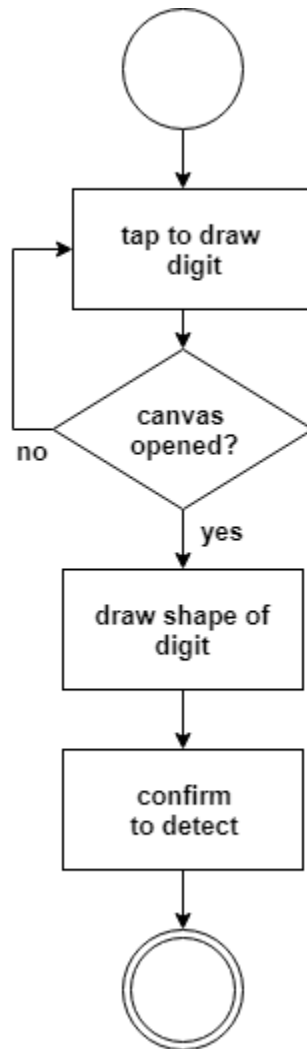


Figure 5: Activity Diagram of Draw Digit Subsystem

Subsystem: View

Recognized digit will be shown as output on English digits and words on this subsystem. There are some cases for which output might not be generated and a failure message will be shown. Following diagram on Figure 6 explains the activity of this subsystem.

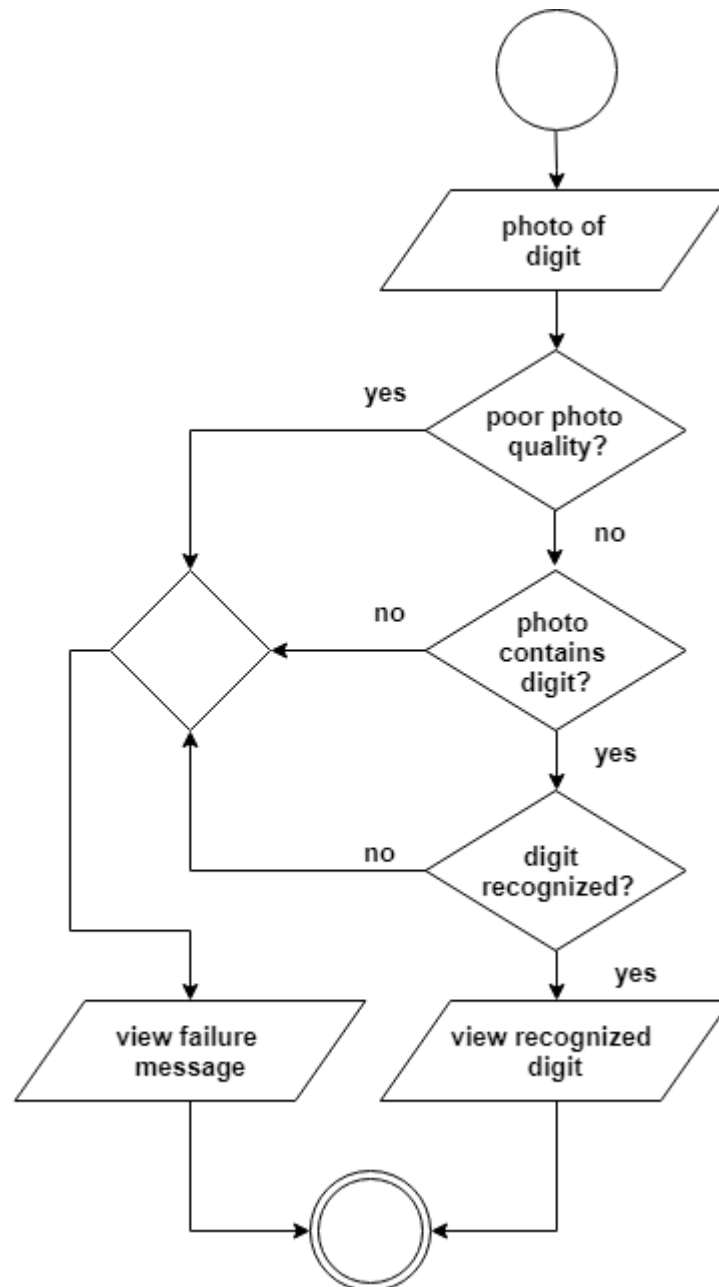


Figure 6: Activity Diagram of View Subsystem

4.4 Conclusion

This chapter concludes here with the scenario based model that includes Use case diagram and Activity diagram. Unlike some scenario based model, Swim lane diagram is excluded here. The technical development of this project does not involve much with the structure given by the Swim lane diagram. Therefore, I hope Use case and Activity diagram will be helpful enough for scenario based model depending on which designing of the app will be developed.

Moreover, other SRS models like- Data model, Class based model and Behavioral model are not necessary for the nature of this project. Hence, these models are also not in the scope of this document.

Chapter Five

Architectural Design

This chapter includes the architectural design steps and diagrams of the project Handwritten Bangla Number Recognizer Android App 'Anko'.

5.1 Introduction

The software architecture of a program is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. Architectural design represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of a system. (2) Architectural design provides big picture of the system. Following figure shows the steps of architectural design-

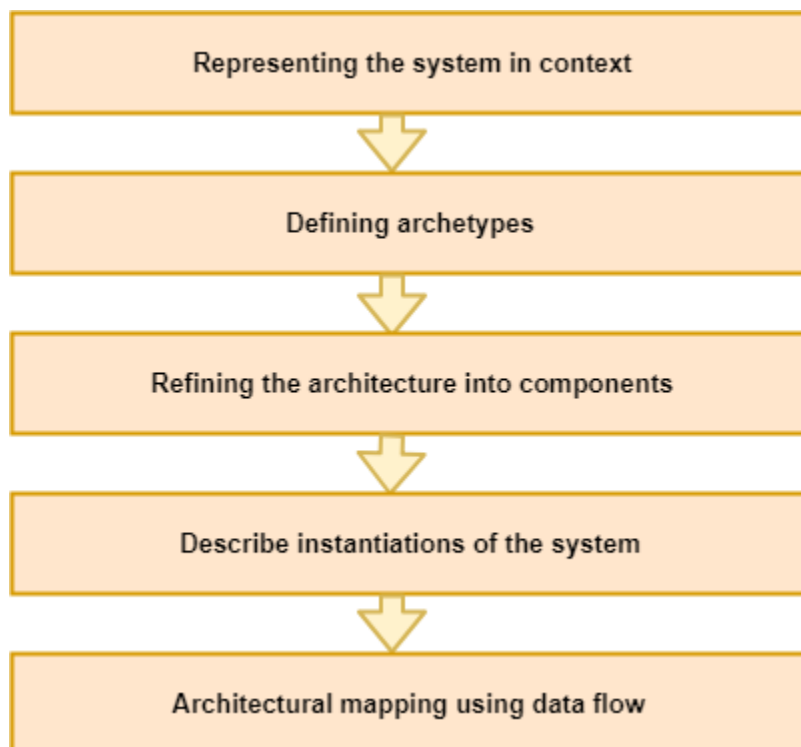


Figure 7: Steps of Architectural Design

5.2 Representing the System in Context

Architectural Context Diagram is used to model the manner in which software interacts with entities external to its boundaries. In architectural context diagram systems that interoperate with the target are represented as-

- Super-ordinate systems: Use target system as part of some higher level processing scheme
- Sub-ordinate systems: Used by target system and provide necessary data or processing
- Peer-level systems: Interact on a peer-to-peer basis with target system to produce or consume data
- Actors: People or devices that interact with target system to produce or consume data

Each of these external entities communicates with the target system through an interface. (2)

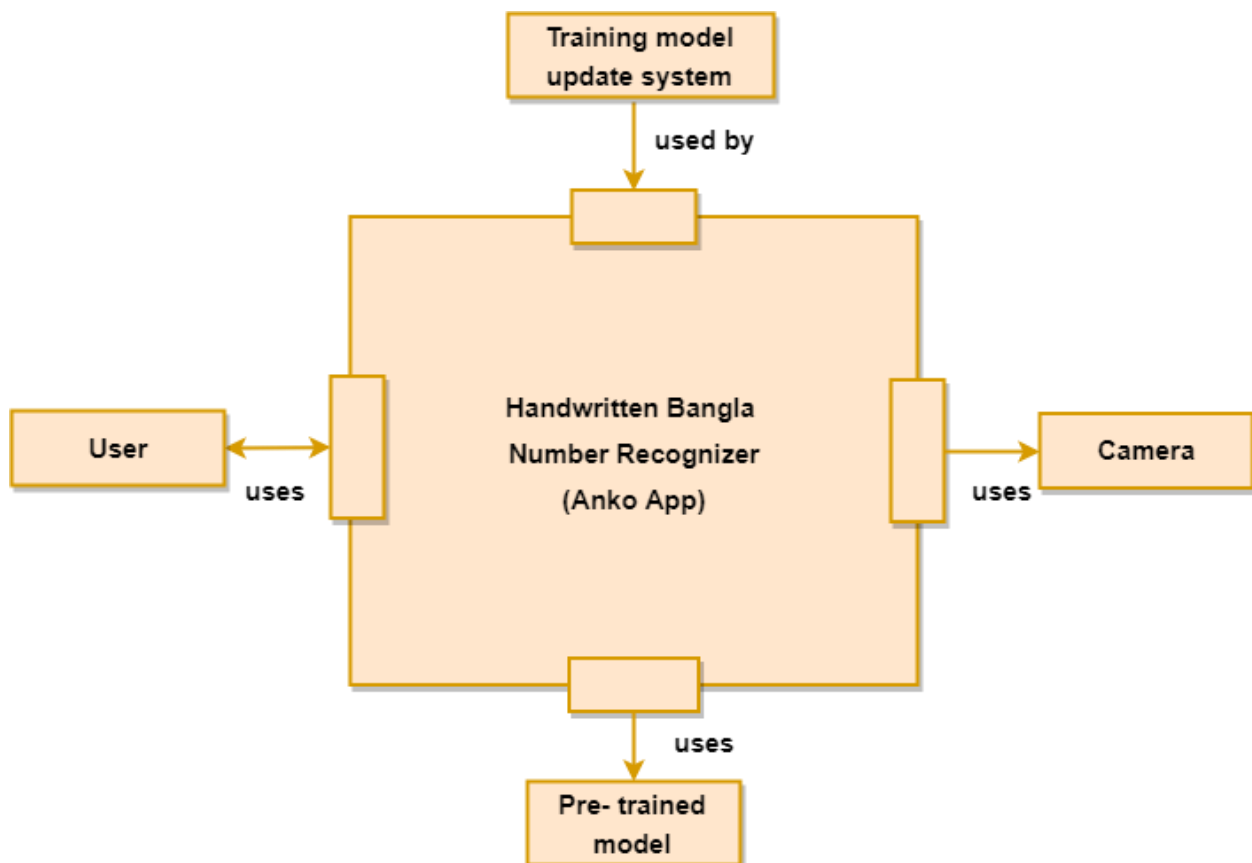


Figure 8: Architectural Context Diagram

Figure 8 shows the architectural context diagram of Handwritten Bangla Number Recognizer Android App. The training model update system is the superordinate system here. Android app users are the actors that are both producers and consumers of information used or produced by the System. The system uses camera of device to take image as input. Pre-trained model is used by the system to recognize digit and is shown as subordinate to it.

5.3 Defining Archetypes

An archetype is a class or pattern that represents a core abstraction that is critical to the design of an architecture for the target system. They represent stable elements of the architecture. Archetypes can be derived by examining the analysis classes defined as part of the requirements model. (2) The archetypes of this System are:

- Image Info
- Classifier
- Classification

The archetypes, their attributes, methods and relationships are illustrated in Figure 9.

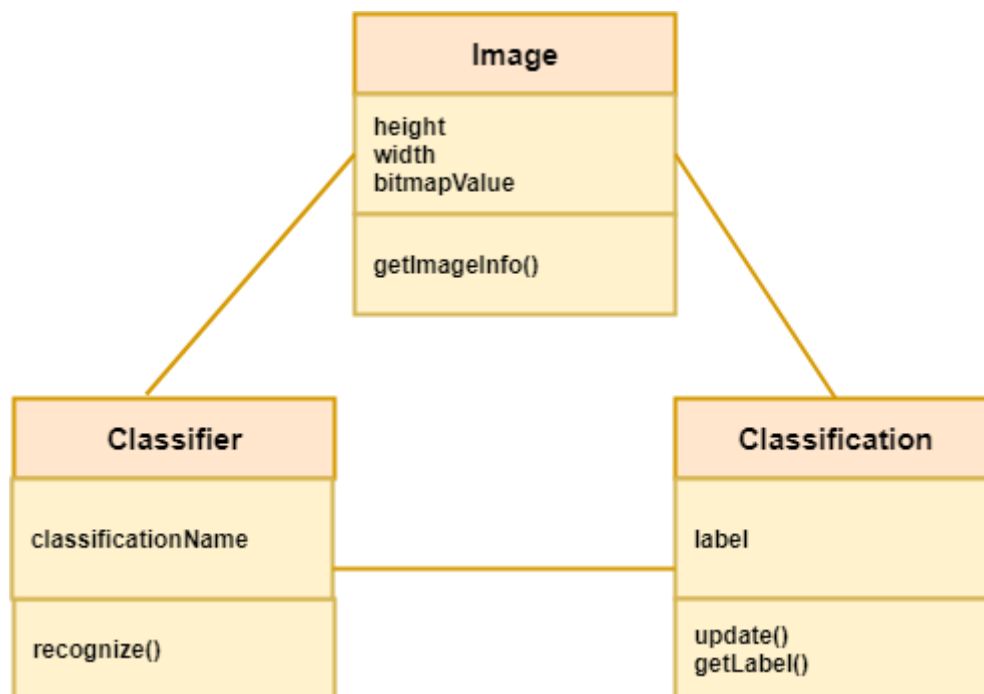


Figure 9: Archetypes

5.4 Refining the Architecture into Components

In this step software architecture is refined into components. These components can be derived from various sources- (2)

- Application Domain provides application components.
- Infrastructure Domain provides design components like design classes.
- The interfaces in the ACD imply one or more specialized components that process the data that flow across the interface.

Anko app consists of the following top-level components-

External communication management: manages tasks related to user input and system output.

Trained model processing: handles training of the dataset related activities.

GUI: manages graphical user interface.

Figure 10 shows overall architectural structure for Anko app with top-level components.

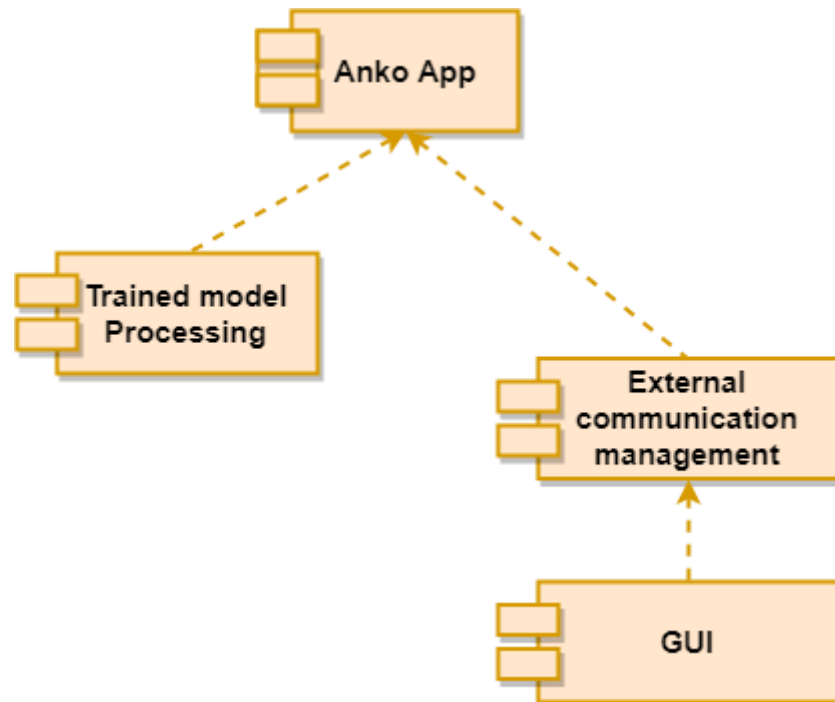


Figure 10: Overall Architectural Structure with Top-Level Components

5.5 Describing Instantiations of the System

Although the major software components have been identified, further refinement is still necessary. To accomplish this, an actual instantiation of the architecture is developed.

Figure 11 illustrates an instantiation of the System architecture. Components shown in Figure 10 are elaborated to show additional detail.

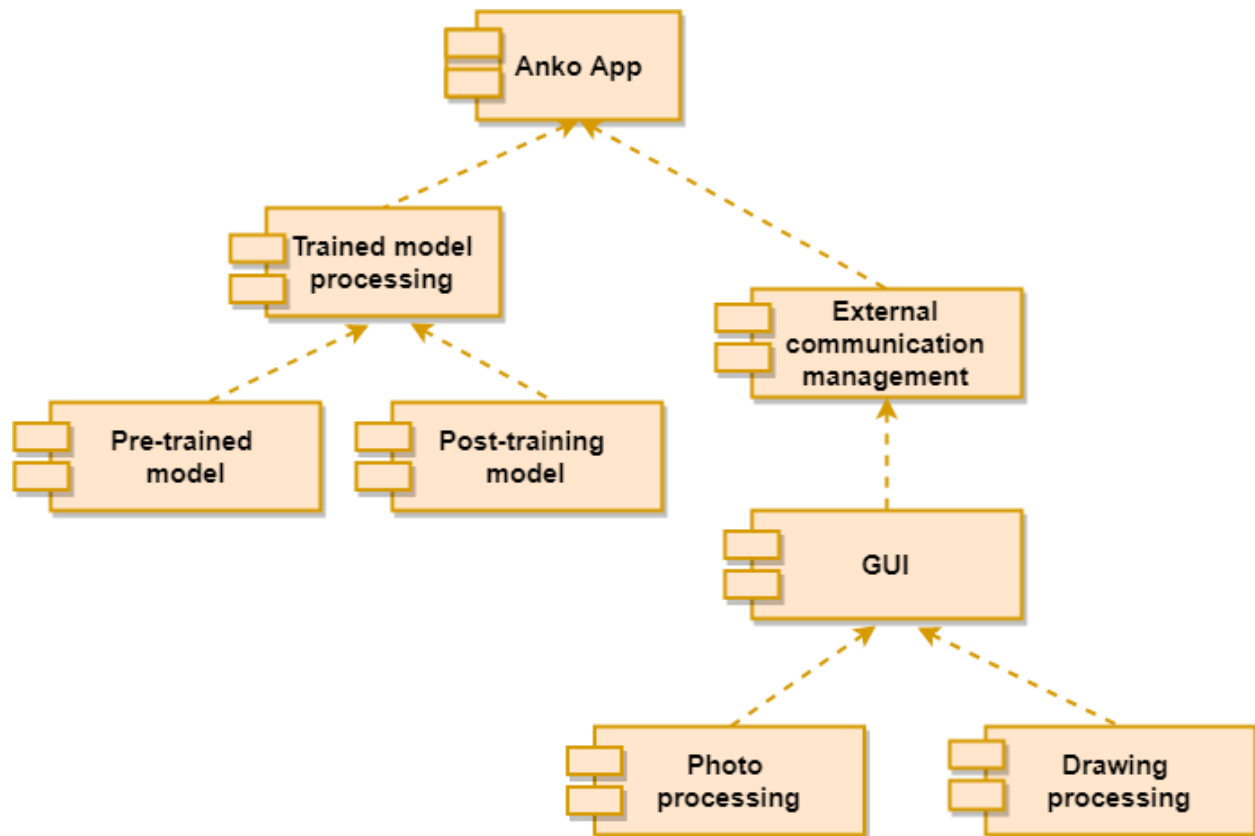


Figure 11: An Instantiation of the Anko App system Architecture

5.6 Architectural Mapping Using Data Flow

A mapping technique called structured design is often characterized as a data flow-oriented design method because it provides a convenient transition from a data flow diagram to software architecture. The transition from information flow to program structure is accomplished as part of a six step process. The level 0 data flow diagram (DFD) of Anko app is shown in Figure 12.



Figure 12: Level 0 Data Flow Diagram

In level 1 data flow diagram, the system is divided into two other subsystems- Training and Recognition. The DFD is illustrated in detail in Figure 13.

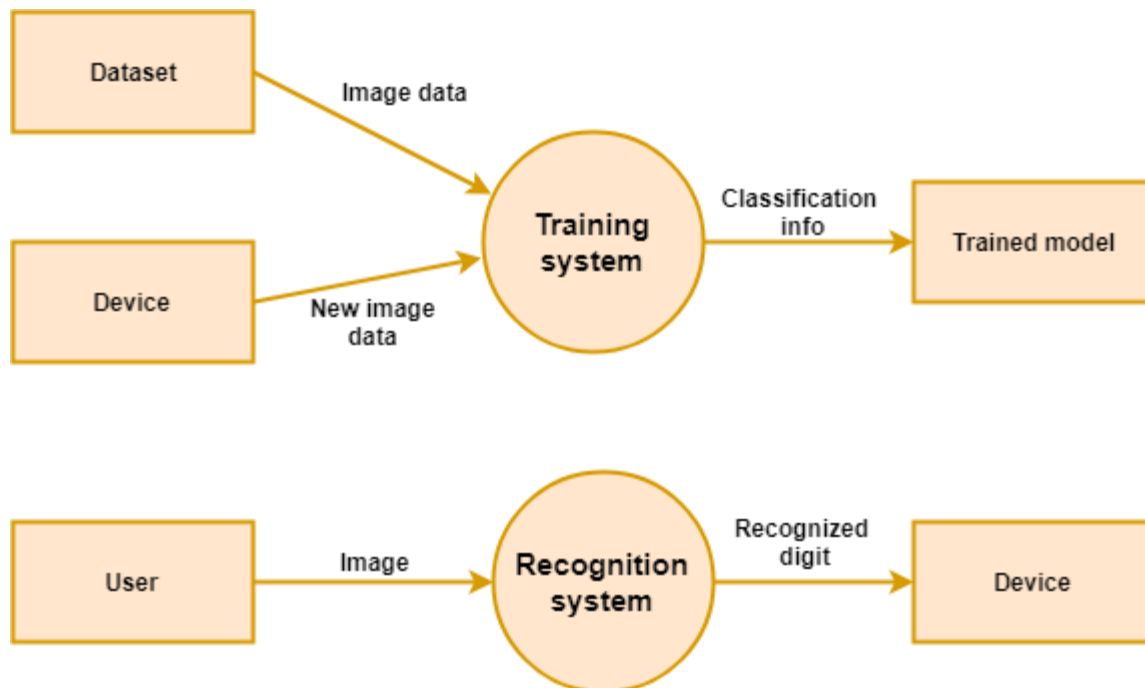


Figure 13: Level 1 Data Flow Diagram

Recognition system of level 1 data flow diagram is shown in detail in Figure 14.

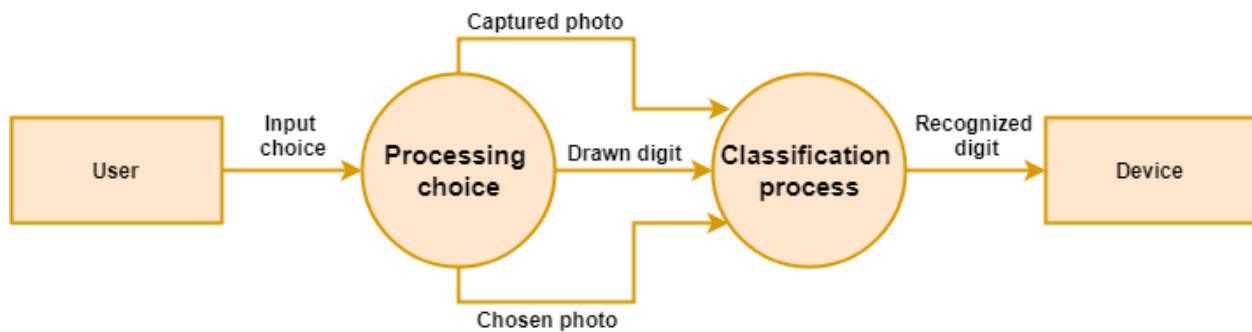


Figure 14: Data Flow Diagram of Recognition process

5.7 Conclusion

The architectural design described on above chapters followed all the steps required for architectural design type. There are other parts of software design- Component level design and User interface design. These are not required for a small scale app project like this which involves more of deep learning than software development. Therefore, these design steps are skipped hoping that Architectural design will be enough for this software development.

Chapter Six

Conclusion

I am very pleased to successfully draw the conclusion of software requirement specification and design documentation of the software project - Anko, A Handwritten Bangla Number Recognizer App.

I hope this report will be able to convey a clear picture of the system to all stakeholders, and act as a basis throughout full development cycle. I have tried my best to make a complete, practical, consistent and unambiguous specifications; which helped me tremendously in my understanding of the scope and detailed process of software requirement engineering process. I think that this report has been written in an easy-to-read way as well as with full information required to have a good concept over the idea. I sincerely hope this document will be able to satisfy the goals all stakeholders expect from it. I hope that any reader going through this document can easily understand the whole idea behind the Anko App. Hopefully, it will be an easy path-showing document for the implementation of the application.

Appendix

References

1. Handwritten Bangla Digit Recognition Using Deep Learning By Md Zahangir Alom, Paheding Sidike, Tarek M. Taha
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman, 7th Edition, last accessed on 20.09.18
3. http://www.tutorialspoint.com/software_engineering/software_requirements.htm, last accessed on 22.09.18