INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

Automated design violations for iOS applications

A dissertation by

Miss. Chavika Kodithuwakku

2018468/w1742135

Supervised by

Mr. Asanjay Fernando

Submitted in partial fulfilment of the requirements for the BEng in Software Engineering degree at the University of Westminster.

November 2021

Table of Contents

CHAPTER 1: INTRODUCTION 2

1.1 Chapter Overview 2

1.2 Problem Domain 2

1.3 Problem Definition 3

1.4 Problem Statement 4

1.5 Research Motivation 4

1.6 Existing Work 4

1.7 Research Gap 6

1.8 Contribution to the body of knowledge 6

1.9 Research Challenge 7

1.10 Research Questions 7

1. 11 Research Aim 7

1.12 Project Scope 7

1.12.1 In scope 8

1.12.2 Out of scope 8

1.12.3 Diagram showing prototype feature 8

1.13 Resource Requirements 9

1.13.1 Hardware Resource Requirements 9

1.12.2 Software Resource Requirements 9

1.12.3 Skill Requirements 9

CHAPTER 2 : METHODOLOGY 9

2.1 Chapter Overview 9

2.2 Research Methodology 10

2.3 Development Methodology 10

2.4 Project Management Methodology 11

2.4.1 Schedule using the Gantt chart 11

2.4.2 Deliverables, Milestones, and dates of deliverables 11

2.4.3 Risk Management 12

CHAPTER 1: INTRODUCTION

1.1 Chapter Overview

Visual User Interface (GUI) facilitates user-software application interaction. Users will become annoyed and remove the application if the user interface is badly designed for Android or iOS. Poorly constructed user interfaces are referred to as UI design violations by the industry. Here is a tool for Android GUI consisting of a gallery website for material design guidelines and a tool website (Bo Yang, 2021) (Bo Yang, 2021).

However, these approaches do not have the ability to detect iOS application violations, and neither does Google Material Design mention when these guidelines are used.

An automated method for identifying UI rules contraventions against the don't-use of Material Design and Human Interface Guidelines is developed in this study.

1.2 Problem Domain

App users should be able to intuitively figure out the app's information structure and action flow thanks to the graphical user interface (GUI). Unfortunately, the user interfaces for iOS and Android may be badly designed. Users refer to uninstalling applications with a badly designed user interface. Poorly designed user interfaces are referred to as UI design violations. UI design smells violate good UI design principles in the same way as code smells reveal violations of good software design rules (G. Suryanarayana, November 2014). Design systems for a wide range of goods, platforms, and services are included by UI design guidelines (Jad Limcaco, 2020).

In this paper, we employ Material Design and Human Interface Guidelines as design system case studies. There is a lot more to these guidelines than just UI aesthetics.

Seven main design dimensions are included here: Layout, Typography, Iconography, Navigation, Communication, Color and Form. As well as there are four component design features: Anatomy, Placement, Behavior and Usage. Also, HIG (Human Interface Guideline) contains instructions for iOS applications when using such components, as well as various usages of those components.

The detection of UI design that does not follow the design standards has received little support. Code smell detection and remediation tools have been created (Jad Limcaco, 2020) (Martin Fowler, November 2018) (F. Palomba, 2013) (H. Liu, 2015) and other programming or stylistic problems (loveeclipse, 2020) (Anon., 2020) (Anon., 2020) as well as to identify violations in

Android applications. Some approaches may discover errors between UI mockups and completed UIs (K. Moran, 2018 May) or differentiate unusual designs from regular designs (Anon., 2020) (D. Zhao, 2020) (Z. Wu, 2020).These methods, however, are unable to detect iOS applications that violate a UI design system's visual rule.

1.3 Problem Definition

In our day to day lives, we must spend a lot of time with mobile applications. Even whether you are going someplace to hire a tuk-tuk or purchase a daily item from an online shopping mall, everyone nowadays uses these mobile applications to make their work simpler. A smart phone has also become quite common in the hands of many folks. As a result, many users are becoming more interested in mobile applications. We must consider the user's flexibility in developing these applications, whether on iOS or Android, as well as how to quickly convert the user to the goal. In today's fast-paced society, the designer must be able to provide the user with a smart solution over time. As a result, when the user is directed to this solution, the user interface between the user and the app is created using this user interface. If this interface is poorly designed, it is a failure. As a result, they uninstall applications faster than they install them.

This does not indicate that the design should be done in a single color or a combination of colors. This relates to how to handle various UI components in a way that is attractive to the eyes of the user.

For this, iOS and Android have applied their own guidelines. Designers and developers can create a rich interface by following these principles and guidelines.

Because these principles and guidelines are not followed; unfortunately, a mobile user interface may be poorly designed. According to the alliterate survey (Anon., 2013 December 05) 42% of consumers would uninstall an app with a poorly designed user interface.

If there are any violations in the UI input by the designer or developer, an attempt has been made to show them and create a report based on the material guidelines for android apps to detect these violations. First, the design UI example used to describe the rules differ significantly from the real UI designs to be evaluated, thus they cannot be easily compared to identify the guideline violation. Second, comparing a UI design to design principles requires an examination of multi-modal information, including a wide variety of component information (type, text content, instance count, color and size/position) as well as the actual displaying of text and pictures (Bo Yang, 2021) (Bo Yang, 2021).

Designers may learn about these design concepts from material design and HIG (Human Interface Guidelines) while developing native applications, hybrid and cross-platform apps, and when to use them, as well as how to use guidelines in each user's personal and user objectives. Now that there are so many resources available, it is simple to learn new things and learn mobile development by using creative ways. Because this design is not aimed at all users, many applications now accessible in app stores are built without any guidelines, and it appears that users are moving away from them quicker than they are installing.

HIG explains how to use components, how to apply these principles to different specific personas, and how to understand them. Material design is the process with the fundamental processes used in the creation of components.

1.4 Problem Statement

Find out whether the user interface they are Designing and Developing is in accord with Material design and Human Interface Guidelines for developers and designers who do not have much understanding about the design principles and guidelines of designing iOS applications as a cross-platform or hybrid, as well as designers working on newer user personas. It requires time, especially for the developer, to understand the fundamental rules from the beginning in order to correct their mistakes.

1.5 Research Motivation

There was no one in the industry where I interned who understood UI principles very well, and I had before learnt to create applications by following tutorials but had not paid enough attention to UI principles. Due to these reasons I had to face many problems while working as a front-end developer there. As a result, I had to learn UI principles from the start and create a different effect for it.

Therefore, I considered developing a tool for detecting violations in iOS applications.

1.6 Existing Work

Citation	Advantages	Limitations
(Bo Yang, 2021) (Bo Yang, 2021)	Manual identification of UI design smells is less	In Material Design, as well as other design systems that define visual

	successful than automatic detection, especially when several pieces of information must be merged or component-variant-sensitive rules are in there.	do/don't-guidelines for a library of UI components in a similar vein, support implicit and de-facto rules. To impose visual design principles in code, eliminate visual-design-related code smells and improve GUI API design. Should be integrated with design tools to allow just-in-time UI design scent detection.
(Kevin Moran, 2018)	handles a significant, tough, and very practical problem with amazing efficiency and precision, and is both useful and scalable in the eyes of industrial designers and developers.	Currently, the tool imposes minor constraints on designers creating Sketch mock-ups, the most important of which is that component bounding boxes do not overlap.
(T. Nguyen, May 2018)	These models may be used to search for UI design samples based on user-provided natural language descriptions and to produce professional-looking UI designs from simpler, less elegant design drafts.	The core concept is to create and deploy sophisticated deep learning models based on recurrent neural networks and generative adversarial networks to learn UI design patterns from millions of existing mobile apps. This only for design patterns; no idea about how components work.
(Asif Imran, 2019)	Analyzing a huge number of class files from four big Java open source software projects yielded software metrics and the detection of design smells.	Determine the tool's accuracy and recall in addition, we intend to develop and test the performance of the data mining and Spectral Clustering modules.

(J. Ruiz, 2020)	Model-driven educational environment that use simulation as a learning aid by creating a user interface and the underlying program using conceptual domain and presentation models.	Provide feedback-enriched simulation tools to aid in the understanding of UI and application design

Table 1.1: Existing Works

1.7 Research Gap

There is a tool that may detect violations in apps that are presently being built for Android (applying Material design) native, hybrid, or cross-platform utilizing the UI concept. It is unique in that it may be used not only by designers but also by developers.

There are three different kinds of inputs available (Bo Yang, 2021) (Bo Yang, 2021).

1. Designers create UI mockups.
2. Developers enter GUI builder mockups (like Android studio layout editor)
3. Testing frameworks take screenshots while the application is running (like Android UI Automator)

Here, there is a theoretical gap.

1.8 Contribution to the body of knowledge

This report is just for Android UIs. However, this does not apply to iOS. Because the principles that govern each of its components differ somewhat from those of android. Furthermore, in Material Design, the components used in each personas face are not smooth, making accurate detection difficult without a HIG (Human Interface Guidelines).

To overcome this gap, a domain contribution might be made.

1. Contribution to the Software Engineering domain :
    - This research will be offering a novel way to identify when to use those components principles and guidelines.
2. Contribution to the BoK :

- This research will be used in the domain to the identification of violations from the user interfaces of iOS applications

1.9 Research Challenge

There are differences between MD (Material design) and HIG (Human Interface Guidelines). Although md lists iOS components, identifying violations in iOS UI based on that alone is challenging. Identifying iOS UI must be done in accordance with the guidelines and principles they have established (AppsChopper, May 2021).

Visual design smells are like code smells in source code. We investigate 93 don't-do-that rules from Google's Material Design, a complex design framework (Bo Yang, 2021) (Bo Yang, 2021). And here, this research will go through this iOS code smell and investigate don't-do-that rules from Google's Material Design, and Apple's Human Interface Guidelines.

This tool attempts to highlight UI violations and how to correct them using examples from the guidelines. However, we only provide an example UI that is correctly structured for violations. It is not a comprehensive redesign of the user interface. But here we can fix it. According to the example in the violations, after redesigning the UI, the user will be able to continue interacting as before and go through the flow without interruption (R. Branaghan, November 2011).

1.10 Research Questions

RQ 1: What are the main components that vary the most depending on the user personas of the UIs you are developing or designing?

RQ 2: When developing iOS apps, you pay close attention to Apple's HIG (Human Interface Guidelines) or Google's MD (Material design)?

1. 11 Research Aim

This research paper will guide designers and developers in avoiding UI violations in iOS applications and releasing a quality product to the app store.

1.12 Project Scope

The project objectives listed below will be used to categorize and explain the research objectives.

1.12.1 In scope

- Guide designers and developers in avoiding UI violations in iOS applications and releasing a quality product to the app store.
- go through this iOS code smell and investigate don't-do-that rules from Google's Material Design, and Apple’s Human Interface Guidelines.

1.12.2 Out of scope

- In Material Design, as well as other design systems that define visual don't use and use for a library of UI components in a similar vein, support implicit and de-facto rules.
- To apply visual design standards in code, evaluate visual-design related code smells and improve GUI API design.
- Include design tools to enable the identification of UI design smells right before they happen.

1.12.3 Diagram showing prototype feature

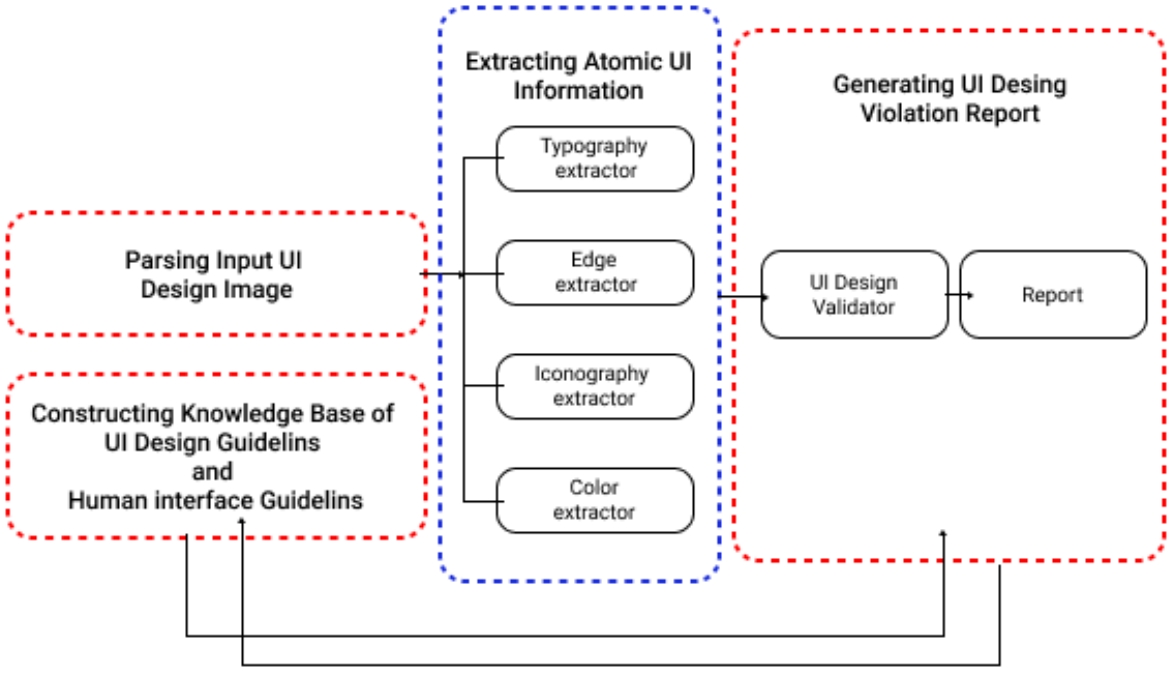


Figure 1.1: Diagram showing prototype feature

1.13 Resource Requirements

1.13.1 Hardware Resource Requirements

- Memory 8 GB 3733 MHz - To handle large datasets.
- Disk space - Store the dataset.
- Processor 1.1 GHz Dual-Core Intel Core i3 - A process with multi-threaded performance is good for compiling, data analysis, model training and data science related tasks.

1.12.2 Software Resource Requirements

- Python - lightweight programming language with a simple syntax
- Google Colab - Make use of server computing power
- PyCharm - A powerful IDE for Python
- Draw.io - For creating graphs and infographics
- Mendeley - Citation and reference management software
- Microsoft Office / Google Docs - Use to manage document submissions
- GitHub - Manage your source code
- Google forms - To collect evaluator responses

1.12.3 Skill Requirements

- Most people have a basic understanding of machine learning algorithms.
- Understanding of linear regression in statistics
- Expertise in assessing machine learning-based methods.
- Skills in writing for a research paper.

CHAPTER 2 : METHODOLOGY

2.1 Chapter Overview

This chapter discusses how to begin, plan, and carry out research, as well as the optimal choice for each category under Saunder's Research Onion. This section focuses on the most efficient approach to manage research utilizing best practices, methodologies, and processes all the way to completion.

2.2 Research Methodology

Research philosophy	For the sake of this essay, this will be employed since it favors quantifiable approaches like surveys, observations, and experiments. The interviews with domain experts and quantitative evaluation of the existing studies and evaluation of the final product and portrayal of correctness using numerical evidence are therefore crucial for this project.
Research approach	Here, deductive reasoning is employed since testing and assessing the hypothesis will show that the conclusion is correct. The proposed solution will be created and then tested to ensure it is accurate.
Research strategies	The research strategy is the approach you use in order to get answers to the research questions. Interviewing domain experts will be a key technique in this study. Surveys may also be utilized with a small group of domain experts.
Research choice	It's your choice whether to limit the scope of the study to a single methodology. Most of the study will be conducted using quantitative approaches, as mentioned in the research philosophy section.
Time zone	In a cross-sectional research, data is gathered over a period, and participants are chosen based on an observable variable. This study started with data collected from a cross-sectional design.

Table 2.1: Research Methodology

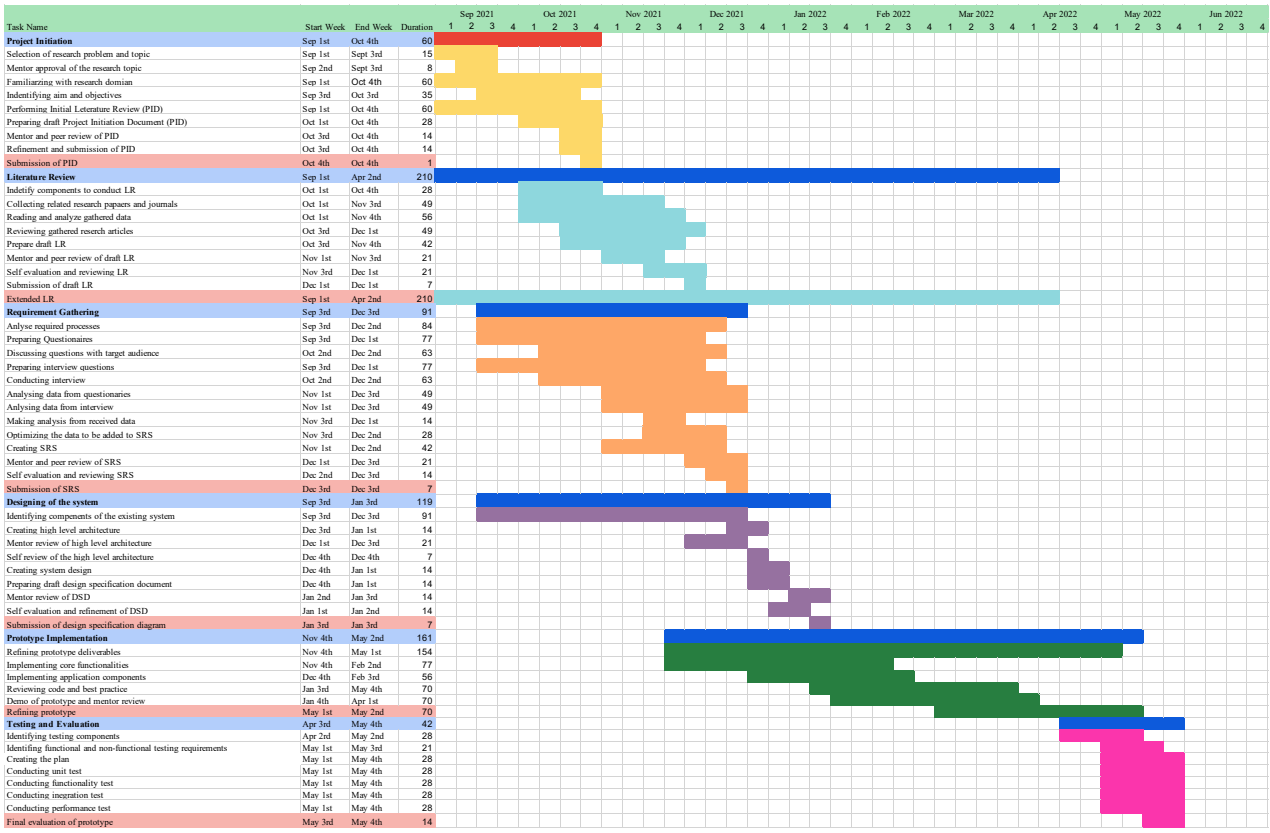
2.3 Development Methodology

There are several methods to structure the development flow process.
Because we will be presenting a prototype of our intended solution at the conclusion of the submission, the prototyping model will be employed here for the development approach. Prototyping model participants will make critical decisions. Developers will actively participate in requirement collecting, and it will be viewed as a collaborative effort between requirement sources and the developer.
As a result, the Prototyping approach will be better appropriate for this study.

2.4 Project Management Methodology

This will be chosen as the project management approach in this case due to its capacity to meet deadlines on time while allowing for strong management abilities and a process-driven strategy.

2.4.1 Schedule using the Gantt chart



Task Name	Start Week	End Week	Duration
Project Initiation	Sep 1st	Oct 4th	60
Selection of research problem and topic	Sep 1st	Sept 3rd	15
Mentor approval of the research topic	Sep 2nd	Sept 3rd	8
Familiarzing with research domian	Sep 1st	Oct 4th	60
Indentifying aim and objectives	Sep 3rd	Oct 3rd	35
Performing Initial Leterature Review (PID)	Sep 1st	Oct 4th	60
Preparing draft Project Initiation Document (PID)	Oct 1st	Oct 4th	28
Mentor and peer review of PID	Oct 3rd	Oct 4th	14
Refinement and submission of PID	Oct 3rd	Oct 4th	14
Submission of PID	Oct 4th	Oct 4th	1
Literature Review	Sep 1st	Apr 2nd	210
Indetify components to conduct LR	Oct 1st	Oct 4th	28
Collecting related research papaers and journals	Oct 1st	Nov 3rd	49
Reading and analyze gathered data	Oct 1st	Nov 4th	56
Reviewing gathered reserch articles	Oct 3rd	Dec 1st	49
Prepare draft LR	Oct 3rd	Nov 4th	42
Mentor and peer review of draft LR	Nov 1st	Nov 3rd	21
Self evaluation and reviewing LR	Nov 3rd	Dec 1st	21
Submission of draft LR	Dec 1st	Dec 1st	7
Extended LR	Sep 1st	Apr 2nd	210
Requirement Gathering	Sep 3rd	Dec 3rd	91
Anlyse required processes	Sep 3rd	Dec 2nd	84
Preparing Questionaires	Sep 3rd	Dec 1st	77
Discussing questions with target audience	Oct 2nd	Dec 2nd	63
Preparing interview questions	Sep 3rd	Dec 1st	77
Conducting interview	Oct 2nd	Dec 2nd	63
Analysing data from questionaries	Nov 1st	Dec 3rd	49
Anlysing data from interview	Nov 1st	Dec 3rd	49
Making analysis from received data	Nov 3rd	Dec 1st	14
Optimizing the data to be added to SRS	Nov 3rd	Dec 2nd	28
Creating SRS	Nov 1st	Dec 2nd	42
Mentor and peer review of SRS	Dec 1st	Dec 3rd	21
Self evaluation and reviewing SRS	Dec 2nd	Dec 3rd	14
Submission of SRS	Dec 3rd	Dec 3rd	7
Designing of the system	Sep 3rd	Jan 3rd	119
Identifying compenents of the existing system	Sep 3rd	Dec 3rd	91
Creating high level architecture	Dec 3rd	Jan 1st	14
Mentor review of high level architecture	Dec 1st	Dec 3rd	21
Self review of the high level architecture	Dec 4th	Dec 4th	7
Creating system design	Dec 4th	Jan 1st	14
Preparing draft design specification document	Dec 4th	Jan 1st	14
Mentor review of DSD	Jan 2nd	Jan 3rd	14
Self evaluation and refinement of DSD	Jan 1st	Jan 2nd	14
Submission of design specification diagram	Jan 3rd	Jan 3rd	7
Prototype Implementation	Nov 4th	May 2nd	161
Refining prototype deliverables	Nov 4th	May 1st	154
Implementing core functionalities	Nov 4th	Feb 2nd	77
Implementing application components	Dec 4th	Feb 3rd	56
Reviewing code and best practice	Jan 3rd	May 4th	70
Demo of prototype and mentor review	Jan 4th	Apr 1st	70
Refining prototype	May 1st	May 2nd	70
Testing and Evaluation	Apr 3rd	May 4th	42
Identifying testing components	Apr 2nd	May 2nd	28
Identifing functional and non-functional testing requirements	May 1st	May 3rd	21
Creating the plan	May 1st	May 4th	28
Conducting unit test	May 1st	May 4th	28
Conducting functionality test	May 1st	May 4th	28
Conducting inegration test	May 1st	May 4th	28
Conducting performance test	May 1st	May 4th	28
Final evaluation of prototype	May 3rd	May 4th	14

Figure 2.4.1: Schedule using the Gantt chart

2.4.2 Deliverables, Milestones, and dates of deliverables

Deliverables	**Date**
Presentation of Project Idea and Feedback from Supervisor	27th Aug 2021
Submission of Draft Project Proposal (Draft PP)	02nd Sep 2021
Evaluation of Draft Project Proposal (Presentations and Feedback)	21st Sep 2021
Submission of Literature Review	21st Oct 2021

Submission of Project Proposal (Final PP)	4th Nov 2021
Ethics forms	4th Nov 2021
Submission of SRS	25th Nov 2021
Presentation of Implementation and Proof of Concept Version 1 and Feedback from Supervisor	17th Dec 2021
Interim Progress Report (IPR)	27th Jan 2022
Presentation of Implementation and Software Demo and Feedback	4th March 2022
Test and Evaluation Report	17th Mar 2022
Submission of Draft Project Reports	31st Mar 2022
Feedback on Report (Feedback) by Supervisor	8th Apr 2022
Final Project Report	10th Apr 2022
Viva Voce Examination	5th May 2022

Table 2.3.2: Milestone and dates

2.4.3 Risk Management

There are several sorts of project risks to be concerned about. If risks are not identified and managed properly, they can lead to cost, scheduling, or performance issues. Strategic risks (choosing the wrong technology, which will generate future risk), operation risks (caused by bad implementation process issues), and risks connected with external variables such as natural disasters, pandemics, and so on will be the major categories of risks associated with this project.

Risk	**Severity**	**Mitigation**
Unexpected resource failures	High	Use source code management tools and save documents on Google Drive.
Insufficient hardware or lack	High to Moderate	Instead of using a local machine, use the computing capacity of a server.
Limited availability of project evaluators	Moderate to high	Especially important in the context evaluators ahead of time, and if they are unavailable, contact industry specialists.

Limited knowledge in key areas	High	Conduct self-study/experiments, get input from experts and search about other pathways.

Table 2.3.3: Risk management