



INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

**UNIVERSITY OF WESTMINSTER (UOW) BEng/BEng. (Hons) in
Software Engineering**

Final year Project 2019/2020

Project Initiation Document

For

Workflow and Activity-based automated user interface generator
approach for mobile application development low code platforms

By

2014062 – W1583051

M.K. Sahan Thinusha Wijayananda

Supervised by

Ms Minoli De Silva

Table of Contents

1.0 Introduction.....	3
1.1 Background to the problem.....	3
1.2 Problem Statement	4
2.0 Problem and Motivation	4
2.1 Research Question	4
2.2 Project Aim	5
2.3 Motivation.....	5
2.4 Research Objectives	5
3.0 Literature Review.....	5
4.0 Prototype	7
4.1 Rich Picture.....	7
5.0 Methodology	7
5.1 Development Methodology	7
5.2 LESP	8
6.0 Functional / Non functional requirements	8
8.0 Resource Requirements	9
8.1 Software Requirements	9
8.2 Hardware Requirement	9
9.0 Reflection on the contribution.....	10
10.0 Time Scale	10
11.0 Reference	11

List of Tables

Table 1: Research Objectives.....	5
Table 2: LESP	8
Table 3: Functional Requirements	8
Table 4:Non-functional Requirements.....	8
Table 5 : Software Requirements.....	9

Table of Figures

Figure 1: Rich Picture	7
Figure 2: Time Scale	10

1.0 Introduction

Developing a mobile application using a programming language from scratch is complex and more time-consuming. What if a user can build their own desired applications along with the User Interface(UI) from analyzing the inputs.

1.1 Background to the problem

Mobile applications or apps are software applications that are designed to run on Smartphones, tablet computers, and other handheld devices(Brendza and Howard. ,2013). Mobile app development is highly increasing because of the smartphones are widely popular among the end-users (Dehlinger and Dixon,2011). The majority of the mobile operating system can be divided into iOS and Android. Mobile apps can put into the following categories: native, web, and hybrid (Masi, Mesbah, and Kruchten, 2013).Mobile apps can be downloaded through the distribution platforms; in other words, app stores that are operated by the owner of the mobile OS (Brendza and Howard. ,2013).

Mobile application development is challenging because of the mobile app development moving forward into the fragmentation rather than unification. Developers face fragmentation issues in fragmentation within the platform and across the platforms (Erfani., Mesbah. and Kruchten.,2013) when developing mobile applications.Sometimes development time effort and budget are multiplied when developing mobile applications.

A low-code(LC) platform creates an application software using a Graphical User Interface (GUI) and some configuration(AlisonDeNiscoRayome,2018). This approach allows those who are not having a software development background to build and test applications. The apps built with low-code (LC) platforms have so many advantages.Those help to speed development, and faster development reduces costs, saves time, and gives high productivity (Totterdale, R. 2018).

Existing low-code platforms are using drag and drop approach for UI designing.For example “Mendix” the current leader of the low-code platform using drag and drop editor for the UI implementation part but if the user doesn't have much knowledge about the UI designing the

app which is going to be generated by the platform doesn't meet the proper mobile UI/UX standards.

1.2 Problem Statement

Companies, individuals who are planned to develop an application, and developers face challenges when they are planned or developing a mobile application. Clients or companies face budget issues, take too much time to build a solution.

Developers also facing challenges when they are developing applications in practice, such as developing across multiple platforms, code reusing problems for the separate applications, take time to develop an application from scratch, user interface(UI) designing and connecting each of them. Also, they face issues with the development testing phase; they have to write test scripts, or they have to test manually what they are developed (Masi, Mesbah, and Kruchten, 2013). Another problem is mobile application projects change rapidly. Also, developers say “changing the application requirements means changing UI/logic after that, test scripts must be constantly rewritten.” (Masi, Mesbah, and Kruchten, 2013).

An individual who is running a small business but does not have a budget for hiring a developer or does not have time to develop an application for his business should have a way to develop or generate a mobile application by themselves. Also, a business person doesn't have any understanding of mobile UI/UX designing to represent the layout.

Current existing mobile application generator comes with drag and drop UI designer tool for design UI, but the user must have a knowledge about UI designing to generate an application using those platforms.

2.0 Problem and Motivation

2.1 Research Question

How to generate cross-platform mobile application processing standardized modelling language.

How to generate a User Interface using Machine Learning techniques.

2.2 Project Aim

The main aim of this project is to develop a framework that can generate a cross-platform mobile solution generator along with an automated generation of User Interface for a specific domain by processing a given workflow diagram.

2.3 Motivation

In mobile application development, lot of time and need a skilled developer to build the application and maintain it. Creating a mobile application in the traditional way of development and testing is a huge cost for the business. If the user used a current low code platform to generate the application, he must have a knowledge about UI designing.

2.4 Research Objectives

Objective	
1	Identifying the most suitable technique for getting the requirements Research has to be carried out to find out the most suitable technique for getting software requirements from the user to generate a suitable UI.
2	Developing an ML model In order to develop the proposed system, Research should be done for how the ML model can be developed, which can identify user requirements and logic which need to generate the mobile solution.
3	Defining the parameters for the ML model Research should be carried out on how the parameters for the ML model should be defined in order to get accurate results.

Table 1: Research Objectives

3.0 Literature Review

Brendza, M., and Howard, B. (2013) proposed a mobile app generator by processing an application description using the provider-specific app information. The author suggested the generation of the mobile application by processing the device, comprising the previously generated base-application code. This proposed way support creating, modifying, and publishing the mobile application. User Interfaces (UI) is generated by processing provider-specific contents. Based on the research, this app builder generate UI by previous and based on the deploying device only, and it's not acquiring any data from the user.

Acerbis et al. (2015) proposed a solution for quick development to the web and mobile application developments. They propose a comprehensive framework called WebRatio platform to develop mobile applications using model-driven architecture. The system using Interaction Flow Modeling Language (IFML), which is a standardized modelling language. The tool is featuring model checking and code generation, which can publish web and cross-platform mobile applications. My point is this proposed approach is good if the generated application is taking more native application behaviours then this one can be used to hardware-specific application generation also.

Bajwa, I., Samad, A. and Mumtaz, S. (2009) proposed a Natural Language Processing (NLP) Object Orientation (OO) software modelling system. A user writes the requirements in simple English in paragraphs, and the system process those paragraphs using NLP, and that generates the classes, objects, and related attributes and methods. Then UML diagrams are also generated based on the requirements. This solution not generated any source code or User Interface, but the good approach from NLP.

LACHGAR, M. and ABDALI, A. (2015) proposed a solution which can generate native code for the iOS application using MDA (Model Driven Architecture) approach. For that, they were using Domain Specific Language (DSL) to abstract the low-level boilerplate code to increase the productivity of software engineers. The downstage is this only covered iOS part and required native coding knowledge for the app generation.

Moore, J., Stefaniak, J. and Sulgrove, L. (2003) is proposed a solution that generates the UML diagram to Java source code which is able to identify the packages sub-packages from the diagram and convert into the source code. This is created by parsing class names, class attributes, field names, and other information from JAVA Source files. This only covers Java source code generation part it doesn't provide a way to generate the test cases and UI generation based on the diagram.

Majchrzak, T. and Ernsting, J. (2015) are proposed a solution that generates business cross-mobile applications using MD² architecture. MD² is a model-driven framework for cross-platform development of mobile applications. They are providing code model for the code generator, and it generates the backend and the mobile application based on the model. But this model only suitable for developing business applications it cannot be used to create applications like games.

4.0 Prototype

4.1 Rich Picture

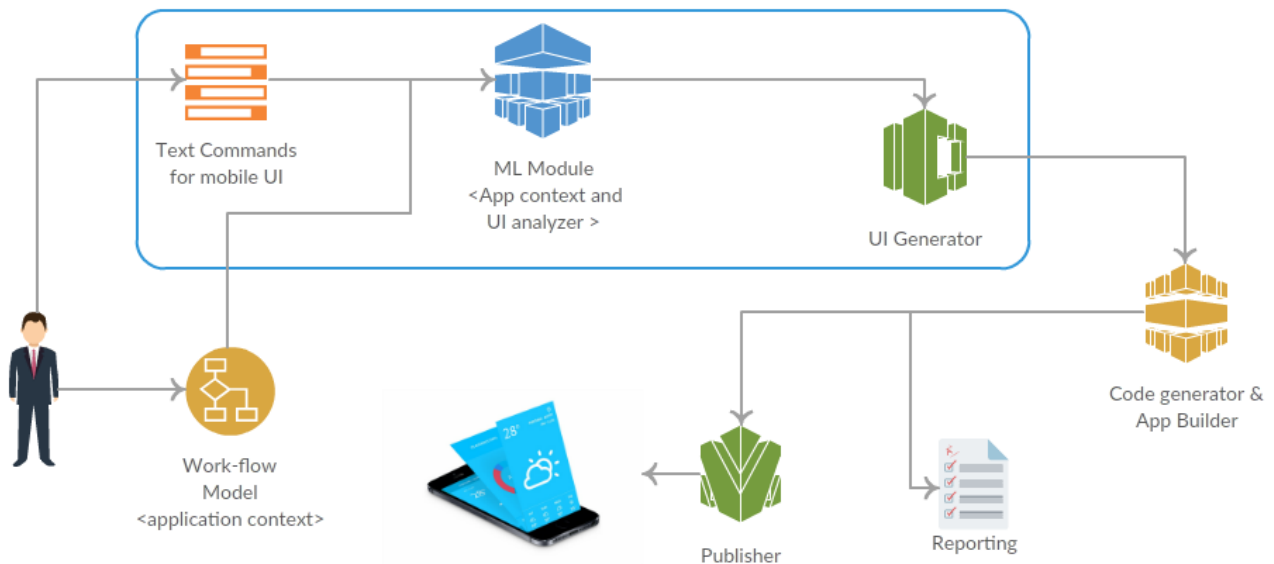


Figure 1: Rich Picture

User has to give the workflow and text commands as an input for the platform. The ML Module will process the input and based on that, it will generate the User Interface. The Code generator processes the workflow command and generates necessary logics and flow for the mobile application. Through the publisher, the user can publish to the Appstore and the reporting section includes report generation and test reports.

5.0 Methodology

5.1 Development Methodology

This kind of project requirement is frequently changed, so the waterfall model is not suitable for this project because the waterfall is not supported for the changing of requirements.

The spiral methodology is used to develop under risk management, and that includes the iterative development process. But the spiral model is used by large expensive and complicated projects which take more than 6 months to 2 years. So this project is a time-limited project and doesn't suit with the spiral methodology (Bookmark, 1988).

Agile methodology having different methodologies with specific steps to produce software by using extreme programming, scrum methods and its very clear. Normally agile methods

are shorter to 2 to 4 weeks, so the development process can be divided into small parts and can get the excellent outcome (Singh, 2008).

5.2 LESP

Legal	Ethical
This project doesn't meet any legal issues because the inputs are getting from the customer.	Data Protection will be not a problem using public data sets.
Social	Professional
This project not subject to any social problems.	The project doesn't affect any professional issues.

Table 2: LESP

6.0 Functional / Non functional requirements

Functional Requirements
Able to design/import workflow, activity diagram
Generate bug-free responsive native look mobile application
Generate automated UI based on the user preference
Generate test cases based on the inputs
Appstore publish feature

Table 3: Functional Requirements

Non-functional Requirements
Recoverability – Data should be recoverable if something went wrong to the system
Security – Security should be improved because users business logics are saved in the system
Performance -the system should be able to respond within the minimum response time.

Table 4:Non-functional Requirements

8.0 Resource Requirements

8.1 Software Requirements

Objective	
1	The operating system Unix-like (macOS Catalina, Ubuntu), Windows 10
2	TensorFlow Machine learning framework
3	Git For software version controlling.
4	Docker virtualization to deliver the system.
5	Google Chrome To run the web application.
6	Documentation tools Microsoft Word, Google Sheets, and Forms To document and manage the project-related documents
7	Intelli IdeaJ Write and manage source code.

Table 5 : Software Requirements

8.2 Hardware Requirement

Core i5 2.1 GHz or higher processor

8GB of minimum RAM

Mobile Device Android/iOS to test the system

9.0 Reflection on the contribution

The challenge is to generate the User Interface analysing the user inputs. And also be able to build the learning model for the UI generation.

The main contribution of the research is to develop an algorithm to generate a responsive mobile application UI for the low-code platform.

10.0 Time Scale

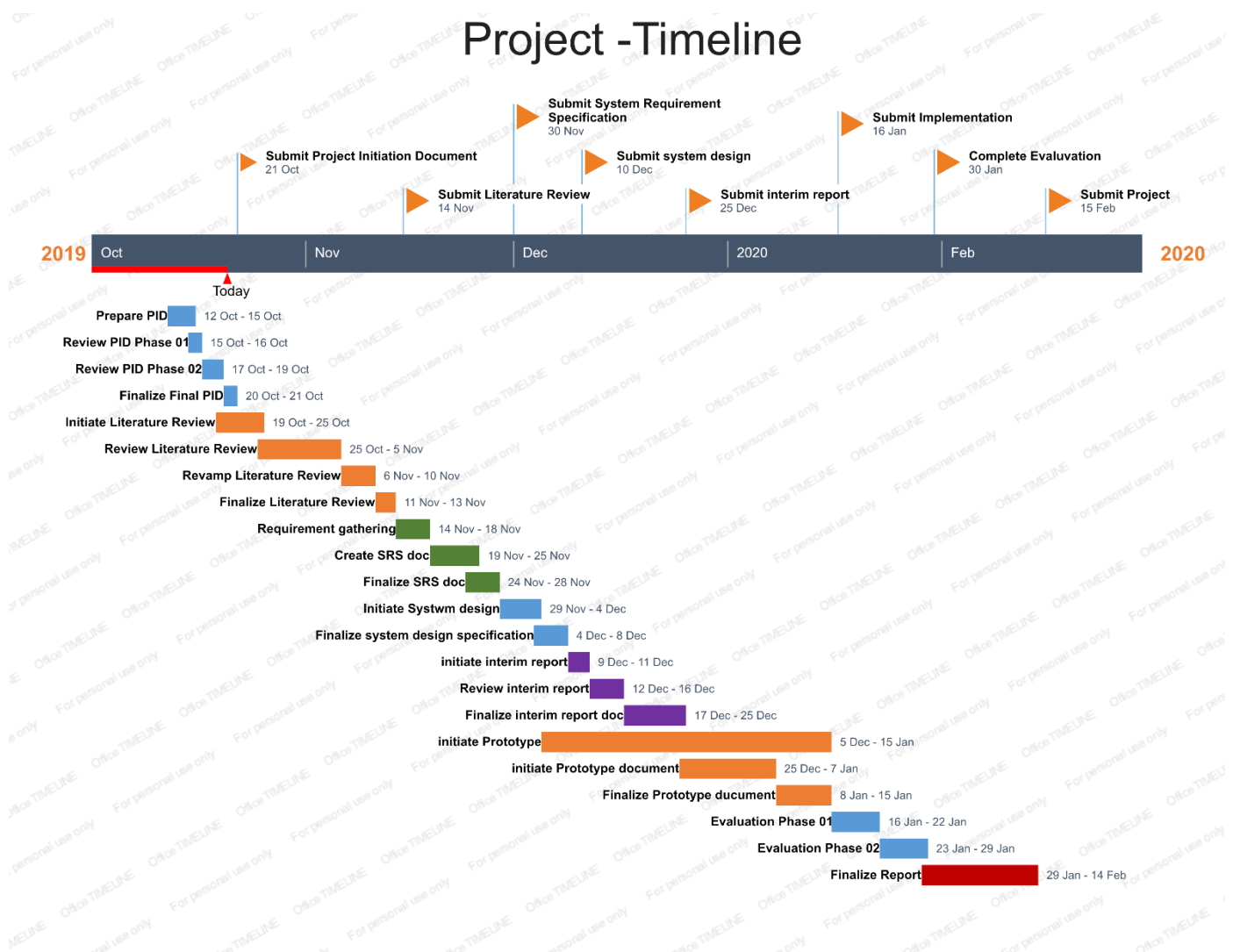


Figure 2: Time Scale


11.0 Reference

1. E. Masi, G. Cantone, M. Mastrofini, G. Calavaro, and P. Subiaco, "Mobile apps development: A framework for technology decision making," in Proceedings of International Conference on Mobile Computing, Applications, and Services., ser. MobiCASE'4, 2012, pp. 64–79.
2. Erfani Joorabchi, M., Mesbah, A. and Kruchten, P. (2013). Real Challenges in Mobile App Development - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/6681334> [Accessed 18 Oct. 2019].
3. Totterdale, R. (2018). [online] Iacis.org. Available at: http://www.iacis.org/iis/2018/2_iis_2018_132-139.pdf [Accessed 19 Oct. 2019].
4. Brendza, M., and Howard, B. (2013). US8656353B2 - Mobile application generator - Google Patents. [online] Patents.google.com. Available at: <https://patents.google.com/patent/US8656353B2/en> [Accessed 20 Oct. 2019].
5. Acerbis, R., Bongio, A., Brambilla, M., and Butti, S. (2015). Model-Driven Development Based on OMG's IFML with WebRatio Web and Mobile Platform. [online] Available at: https://link.springer.com/chapter/10.1007/978-3-319-19890-3_39 [Accessed 20 Oct. 2019].
6. Bajwa, I., Samad, A., and Mumtaz, S. (2009). [online] Pdfs.semanticscholar.org. Available at: <https://pdfs.semanticscholar.org/4e66/8bf145985da6b327fc8238507f5a67960230.pdf> [Accessed 20 Oct. 2019].
7. LACHGAR, M. and ABDALI, A. (2015). DSL and code generator for accelerating iOS apps development - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/7483269> [Accessed 20 Oct. 2019].
8. Moore, J., Stefaniak, J. and Sulgrove, L. (2003). US6560769B1 - Computer-implemented method for generating a UML representation from JAVA source code - Google Patents. [online] Patents.google.com. Available at: <https://patents.google.com/patent/US6560769B1/en> [Accessed 20 Oct. 2019].

9. Majchrzak, T. and Ernsting, J. (2015). Achieving Business Practicability of Model-Driven Cross-Platform Apps. [online] Uia.brage.unit.no. Available at: <https://uia.brage.unit.no/uia-xmlui/handle/11250/2392249> [Accessed 20 Oct. 2019].
10. Bookmark, D. (1988). CSDL | IEEE Computer Society. [online] Computer.org. Available at: <https://www.computer.org/csdl/magazine/co/1988/05/r5061/13rRUwwslzm> [Accessed 20 Oct. 2019].
11. Singh, M. (2008). U-SCRUM: An Agile Methodology for Promoting Usability - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/4599538> [Accessed 20 Oct. 2019].
12. M. Brambilla, P. Fraternali, Interaction Flow Modeling Language – Model-driven UI Engineering of Web and Mobile Apps with IFML. Morgan Kauffman, USA, 2014.
13. J. Dehlinger and J. Dixon, “Mobile application software engineering: Challenges and research directions,” in Proceedings of the Workshop on Mobile Software Engineering. Springer, 2011, pp. 29–32.
14. AlisonDeNiscoRayome. Low-codeplatforms: A cheatsheet. TechRepublic, 2018.
15. M. Brambilla, A. Mauri, E. Umuhzoza, “Extending the Interaction Flow Modeling Language (IFML) for Model Driven Development of Mobile Applications Front End”, MobiWIS 2014, Springer LNCS 8640, pp. 176-191.

Workflow and Activity-based automated user interface generator approach for mobile application development low code platforms

Plagiarism Report



Assignment Inboxpreferences



Welcome to your new class homepage! From the class homepage you can see all your assignments for your class, view additional assignment information, submit your work, and access feedback for your papers.

Hover on any item in the class homepage for more information.

Class Homepage

This is your class homepage. To submit to an assignment click on the "Submit" button to the right of the assignment name. If the Submit button is grayed out, no submissions can be made to the assignment. If resubmissions are allowed the submit button will read "Resubmit" after you make your first submission to the assignment. To view the paper you have submitted, click the "View" button. Once the assignment's post date has passed, you will also be able to view the feedback left on your paper by clicking the "View" button.

Assignment Inbox: MODULE: (2019) 6COSC012C.Y Final Year Project (all courses) (IIT Sri Lanka) _75460_1

Assignment Title	Info	Dates	Similarity	Actions
PID Final Version Submission [FOR CS / SE Only]		Start 04-Oct-2019 7:58AM Due 21-Oct-2019 11:00AM Post 01-Dec-2019 12:00AM	6% 	Resubmit View 