

Automated Front-End View Capturing System for Flutter Based Applications

By

Christian Joseph De Silva Tan
12002015

BACHELOR'S DEGREE
in

Information Technology
Faculty of Engineering and Information Technology



SWISS GERMAN UNIVERSITY
The Prominence Tower
Jalan Jalur Sutera Barat No. 15, Alam Sutera
Tangerang, Banten 15143 - Indonesia

Revision after the Thesis Defense on July 1 2024

STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Christian Joseph De Silva Tan

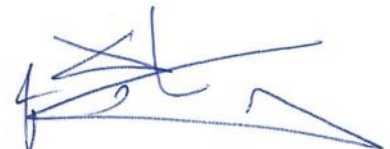
Student


15.7.2024

Approved by:

Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Thesis Advisor


15.7.2024

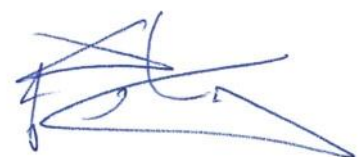
Leonardo Kurnia, S.Kom., B.Eng

Thesis Co-Advisor


15 July 2024

Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Dean


15.7.2024

Christian Joseph De Silva Tan

ABSTRACT

Automated Front-End View Capturing System for Flutter based Applications

By

Christian Joseph De Silva Tan
Dr. Maulahikmah Galinium, S.Kom., M.Sc.
Leonardo Kurnia, S.Kom., B.Eng

SWISS GERMAN UNIVERSITY

This thesis addresses the challenge of streamlining the customer request process for accessing front-end views or pages in applications. The proposed solution is to develop an automated solution that captures views from Flutter framework-based apps. Leveraging PostgreSQL and Python, the prototype generates automated screenshots, which are then displayed on a React.js web page. Docker Containerization ensures seamless deployment on Commsult's staging server. The system architecture involves a Python script running within a Docker container, utilizing Pyppeteer to capture headless screenshots of the Flutter app. These screenshots are stored in Base64 format within a PostgreSQL database, accessible via a React.js interface. The script is then integrated to a web-based customer portal, to allow ease of access to this new feature. While the initial prototype demonstrates feasibility, ongoing development, and production offer opportunities for improvement.

Keywords: Automation, Python Scripting, PostgreSQL, Docker, Front-end Development, Software Tools.

© Copyright 2024
by Christian Joseph De Silva tan
All rights reserved

DEDICATION

I dedicate this research to Commsult AG, the company that has fostered limitless learning opportunities and a commitment to excellence.

ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to God for his unwavering guidance, knowledge, and support anytime I felt lost or had concerns. Through you I can do anything, I can do all things, for its you who gives me strength, nothing is impossible. I want to sincerely thank my family for all their sacrifices to give me the resources and education I needed to complete my bachelor's degree. For showering me with constant love and support especially during my darkest days.

I am thankful to Commsult Indonesia and Commsult AG for giving me the opportunity to work on my thesis within their respective companies. Special shout out to Mr. Leonardo Kurnia, for being an amazing mentor, co-advisor, and friend. Thank you for pushing me above my limits and supporting me throughout this thesis period. I would like to thank the people from Commsult Indonesia, namely Mr. Jesse Young, Mr. Aditya Fidiyanto, Mr. Aldi Prayogi, Mr. Juan Emmanuel, Dave Anderson, and Michaela Hannah, for supporting and providing insights during my thesis period. I would like to give a special mention to Mr. Oliver Henkel, Mr. Tino Hönicke, Mr. Sebastian Minke, Mr. Tobias Schenke, and Tom Bucksch, for accommodating me and my thesis partner during our stay in Germany. Thank you for providing insightful knowledge and improvements that can be done during my thesis period and thank you for making our stay in Germany an amazing one.

I would like to thank my friends inside and outside Swiss German University, namely Nazree, Mia, Sam, James, Brandon, Owen, Nico, and Nicko for always being there for me when I am in times of need and helping me keep in check with my thesis. I would like to thank the people from Lighthouse Berlin, especially Ashish, Aldo, Aldi, Bryan, Kunal, Theresia, and Winni for always reaching out to me and providing me a safe space to release my frustrations. Finally, I would like to thank Janet Gloria, my thesis partner, for always being supportive during our thesis period. Thank you for sharing

your knowledge and wisdom to me whenever I felt lost, and thank you for being an amazing thesis partner, internship partner and schoolmate.

TABLE OF CONTENTS

STATEMENT BY THE AUTHOR.....	2
ABSTRACT.....	3
DEDICATION.....	5
ACKNOWLEDGEMENTS.....	6
TABLE OF CONTENTS.....	8
LIST OF FIGURES	10
LIST OF TABLES.....	12
CHAPTER 1 - INTRODUCTION.....	13
1.1 BACKGROUND	13
1.2 RESEARCH PROBLEMS	14
1.3 RESEARCH OBJECTIVES.....	16
1.4 RESEARCH QUESTIONS.....	16
1.4 HYPOTHESIS	17
1.5 SIGNIFICANCE OF STUDY.....	17
1.6 LIMITATIONS	18
1.7 RESEARCH SCOPE.....	18
CHAPTER 2 - LITERATURE REVIEW.....	20
2.1 THEORETICAL PERSPECTIVES.....	20
2.1.1 Commsult Products.....	21
2.1.2 Software Development Life Cycle (SDLC)	21
2.1.3 Information Technology (IT) Automation.....	23
2.1.4 Scripting.....	24
2.2 RELATED WORKS	25
2.2.1 Advanced String Pattern Matching by Manuel Langer Alpen-Adria-Universität Klagenfurt.....	25
2.2.2 Exploring Regular Expression Usage and Context in Python by Carl Chapman and Kathryn T.Stolee Iowa State University and North Carolina State University.....	25
2.2.3 Flutter for cross-platform App and SDK development by Lukas Dagne Metropolia University of Applied Sciences 01 May 2019 (Dagne and Vuori, 2019)	26
2.2.4 A multi-criteria comparison of mobile applications built with the use of Android and Flutter Software Development Kits based on many criteria. By Damian Galan, Konrad Fisz*, Piotr Kopniak Lublin University of Technology, Nadbystrzycka March 2021 (Galan, isz and Kopniak, 2021).....	27
2.2.5 Web Development with MongoDB and Node.js by Mithun Satheesh, Bruno Joseph D'mello and Jason Korl.....	27
2.2.6 Comparison of Existing Solutions.....	31
CHAPTER 3 – RESEARCH METHODOLOGY	32
3.1 RESEARCH FRAMEWORK	32
3.1.1 Planning and Analysis	33
3.1.2 Design and Architecture	34
3.1.3 Coding and Prototype Development.....	34
3.1.4 Testing.....	35

3.2	ARCHITECTURE DIAGRAM.....	37
3.3	USE CASE DIAGRAM	40
3.4	ACTIVITY DIAGRAM.....	40
3.5	SOFTWARE TESTING.....	42
3.5.1	<i>Integration Tests</i>	42
3.5.2	<i>Functional Tests</i>	42
3.5.3	<i>Acceptance Tests</i>	43
3.6	THESIS PROGRAMMING ENVIRONMENT.....	43
CHAPTER 4 – RESULTS AND DISCUSSIONS		45
4.1	INITIAL EVALUATION	45
4.1.1	<i>Flowchart Overview</i>	48
4.2	PROTOTYPE DEVELOPMENT	48
4.2.1	<i>Prototype One Puppeteer with DCLI Script</i>	49
4.2.2	<i>Prototype Two Capturing Class Name in Flutter Project</i>	51
4.2.3	<i>Prototype Three Capturing Route in Flutter Application</i>	54
4.2.4	<i>Prototype Four Pypeteer Implementation</i>	56
4.2.5	<i>Prototype Five Sending Captured Screenshots to the Database</i>	58
4.3	BACK-END AND FRONT-END DEVELOPMENT.....	61
4.3.1	<i>Back-end Development</i>	61
4.3.2	<i>Front-end Development</i>	63
4.4	IMPLEMENTATION TO DOCKER.....	66
4.4.1	<i>Initial Deployment Stage</i>	67
4.4.2	<i>Docker Ubuntu and Macbook M1 Silicon chip</i>	68
4.4.3	<i>Running Docker Build with BuildX</i>	69
4.5	DOCUMENTATION COMPARISON	76
4.6	DISCUSSION OF FINDINGS.....	77
CHAPTER 5 – CONCLUSIONS AND RECCOMENDATIONS.....		78
4.7	CONCLUSIONS	78
4.8	RECOMMENDATIONS	79
GLOSSARY		80
REFERENCES		82
CURRICULUM VITAE.....		85

LIST OF FIGURES

Figures	Page
<i>Figure 1.1</i>	15
<i>Figure 2. 1 Software Development Life Cycle Diagram</i>	22
<i>Figure 3. 1 SDLC implemented for this Thesis</i>	33
<i>Figure 3. 2 Architecture Diagram Overview</i>	37
<i>Figure 3. 3 Architecture Diagram of Front-End View Capturing System</i>	37
<i>Figure 3. 4 Use Case Diagram</i>	40
<i>Figure 3. 5 Activity Diagram</i>	41
<i>Figure 4. 1 Ideal Flow of Script</i>	47
<i>Figure 4. 2 Flowchart Overview of System</i>	48
<i>Figure 4. 3 Code Snippet of puppeteer_test.dart</i>	49
<i>Figure 4. 4 Login page captured Using Puppeteer</i>	50
<i>Figure 4. 5 extract_class_name function</i>	52
<i>Figure 4. 6 write_to_main_dart Function</i>	53
<i>Figure 4. 7 find_routes_in_dart_files Function</i>	54
<i>Figure 4. 8 Routes Captured from Flutter Application</i>	55
<i>Figure 4. 9 Printing Routes Found</i>	55
<i>Figure 4. 10 Flow of Pyppeteer Function</i>	56
<i>Figure 4. 11 Pyppeteer function Implemented into Script</i>	57
<i>Figure 4. 12 Pyppeteer Function Combined With base64</i>	59
<i>Figure 4. 13 psycopg2 added to the Pyppeteer function</i>	60
<i>Figure 4. 14 Express.js Requests Found in server.js</i>	62
<i>Figure 4. 15 routeList.js</i>	63
<i>Figure 4. 16 Example of Route name and Image data being Displayed</i>	64
<i>Figure 4. 17 imageDisplay.js</i>	65
<i>Figure 4. 18 React Page</i>	66
<i>Figure 4. 19 Dockerizing script or system</i>	66
<i>Figure 4. 20 Initial Dockerfile Development</i>	67

<i>Figure 4. 21 Google chrome Installation Error</i>	<i>68</i>
<i>Figure 4. 22 Successful Docker Image Build.....</i>	<i>69</i>
<i>Figure 4. 23 Docker Architecture</i>	<i>71</i>
<i>Figure 4. 24 Docker Container kind_bhaskara</i>	<i>72</i>
<i>Figure 4. 25 kind_bhaskara, priceless_darwin and pythonscript Containers.....</i>	<i>73</i>
<i>Figure 4. 26 Docker Network testing-network.....</i>	<i>74</i>
<i>Figure 4. 27 Dbeaver Connected to Database</i>	<i>75</i>
<i>Figure 4. 28 Dbeaver Check Captured Pages</i>	<i>75</i>
<i>Figure 4. 29 React-based Web Page.....</i>	<i>76</i>

LIST OF TABLES

Table	Page
<i>Advantages and Disadvantages of each related works and this Thesis.....</i>	28
Comparison between several available tools	31
<i>Programming Language and Version.....</i>	44
Comparison between system and developer	76