



**UNIVERSITY OF THE PHILIPPINES
OPEN UNIVERSITY**

Master of Information Systems

JEANO FREDERICK U. ERMITAÑO

**GO GURO MOBILE APPLICATION BUILT WITH
ANDROID AND FIREBASE**

Thesis/Dissertation Adviser:

Concepcion L. Khan, I

Faculty of Information and Communication Studies

Date of Submission

12 May 2018

Permission is given for the Following people to have access to this thesis/dissertation:

Available to the general public	Yes
Available only after consultation with author/thesis/dissertation adviser	Yes
Available only to those bound by confidentiality agreement	Yes

Student's Signature:

Signature of Thesis/Dissertation Adviser:

UNIVERSITY PERMISSION PAGE

"I hereby grant the University of the Philippines a non-exclusive, worldwide, royalty-free license to reproduce, publish and publicly distribute copies of this thesis or dissertation in whatever form subject to the provisions of applicable laws, the provisions of the UP IRR policy and any contractual obligations, as well as more specific permission marking on the Title Page."

"Specifically, I grant the following rights to the University:

- a) To upload a copy of the work in the theses database of the college/school/institute/ department and in any other databases available on the public internet;*
- b) To publish the work in the college/school/institute /department journal, both in print and electronic or digital format and online; and*
- c) To give open access to above-mentioned work, thus allowing "fair use" of the work in accordance with the provisions of the Intellectual Property Code of the Philippines (Republic Act No. 8293), especially for teaching, scholarly and research purposes."*

Student Name over Signature and Date

ACCEPTANCE PAGE

This thesis titled Go Guro Mobile Application Built With Android and Firebase is hereby accepted by the Faculty of Information and Communication Studies, U.P. Open University, in partial fulfillment of the requirements for the degree Master of Science/Master of Arts/Master of Information Systems/Master of Development Communication.

Members of the Academic Advisory Committee:

_____. Chair. Advisory Committee
(Signature over printed name) (Date)

_____. Member. Advisory Committee
(Signature over printed name) (Date)

_____. Member. Advisory Committee
(Signature over printed name) (Date)

DR. ALEXANDER G. FLOR, Dean,
(Signature over printed name) Faculty of Information and
Communication Studies (Date)

ACKNOWLEDGEMENTS

God, whose unseen guidance has certainly helped create this capstone, is a Person who I would like to thank for everything that has happened to me during the development of this project. When I spent countless, sleepless night writing code and revising the manuscript, I realized how much He has given me. He gave me the strength to endure the process of this master's thesis. Without Him, this would not be possible at all.

This capstone was developed to help students. Therefore, I would like to acknowledge my batchmates. I would also like to thank my classmates from other subjects in UPOU, even though we are not batchmates, I enjoyed the brief time we had while we were enrolled in those subjects. My batchmates and classmates supported each other in our endeavors, especially during IS295a and IS 295b. Even if our interactions were through online means only, I cherished the times when we all helped each other out.

Teachers can also use this application. Hence, I would like to acknowledge the hardworking teachers in UPOU. Not only those who were my professors before, but all teachers working in UPOU. You have brought learning to far away places. Without you, there would be no UPOU. Because of this, I am sincerely grateful to have teachers like you.

Although I forgot your real name, I would like to acknowledge SimCoder, whose Youtube video tutorials greatly helped me learn Android and Firebase development. Your videos gave my development a direction, forwards, and without them the direction of this capstone would otherwise be backwards.

I would also like to thank the company the I work for, for giving me enough money to pay for the ultrawide monitor that I bought. This increased my productivity considerably. I had Android Studio open in one window and a video tutorial beside it. In addition, I would like to thank my co-workers. They always gave me something to laugh about during times when I was depressed.

I also acknowledge my family, whom I miss so much. They are living in Saudi Arabia at the time of writing. I treasured the times you called me on video, despite while I was busy programming.

Lastly, I would like to acknowledge my sweet girlfriend. She supported my endeavor wholeheartedly, even though it meant there would be less time for each other. She has always given me a reason to smile in face of overwhelming hardships. Therefore, I thank you.

ABSTRACT

In universities or colleges, most students have control of their class schedule. There's also instances of irregular students, whose class times are different, depending on what's available during enrolment. Hence, there are times a student needs help with a lesson, making a project, or reviewing for exams, but there are no available professors or classmates. To find help, one must roam the campus and ask other students face-to-face. Often, they will be discouraged to ask because of fear of rejection or that they're too shy to ask.

The proposal is Go Guro, a mobile application for students who need help quick. The tutee selects the subject they need help and with a press of a button, the app instantly searches for nearby people who also have the app. The tutors can choose to accept or decline the tutee's invitation. Once accepted, a map shows the location of both which can be used to find each other. Once they meet in-person, the study session can now be started by the tutor.

Mobile devices using Android 4.0.3 "Ice Cream Sandwich" up to Android 8.0 "Oreo" was targeted. Android Studio was used for development with Java as the programming language. Firebase was used to provide backend services.

Testing resulted in satisfactory remarks from end-users, indicating that it is practical to use in their college. They would also like to see it published on Google Play for every student and teacher to use, regardless of school. The application lessened the user's rejection fears by making the invitations anonymous, meaning the tutee cannot see the name and face of the tutor and vice-versa. Only when the tutor accepts the invite will they see each other's names and faces. Go Guro does not show the tutors who have rejected the invite, but only those who accepted.

TABLE OF CONTENTS

University Permission Page	i
Acceptance Page	ii
Acknowledgements	iii
Abstract	iv
List of Tables	v
List of Figures	vi
Chapter I: The Problem Domain.....	1
A. Statement of the Problem.....	1
B. Background and Objectives of the Project.....	1
C. Significance and Scope of the Project	1
D Documentation of Existence and Seriousness of the Problem	2
Chapter II: Review of Existing Alternatives.....	3
Chapter II: Review of Existing Alternatives.....	4
Chapter III: Approach to Be Taken in This Subject.....	4
A. Theoretical Framework.....	4
B. Rationale for the Framework	4
C. Technologies You Plan to Consider to Use	5
Chapter IV: Project Plan	6
A. Concept.....	6
B. Methods	8
C. Plan for User Testing and Project Assessment	10
Chapter V: Results and Discussion	12
A. System Usability Scale Results	12
B. System Usability Scale Results Interpretation	13
Chapter VI: Conclusions	16
Chapter VI: Conclusions	17
Chapter VII: Recommendations	18
A. Database Remodel.....	19
B. iOS Port	20
Chapter VIII: References.....	21
Chapter VIV: Appendices.....	22
A. Deliverables and Milestones.....	23
B. Budget.....	25
C. Qualifications	26
D. Contributors / Collaborators	27
E. Resources	29
F. Complete Program Listing	30
G. Technical Reference	31
H. User Manual.....	46
I. Software Requirements Specification	

LIST OF TABLES

Table 1. Point value of each option in System Usability Scale	13
Table 2. Score computation for one respondent.....	14
Table 3. Initial project expenses.....	24
Table 4. Recurring project expenses per month.....	24

LIST OF FIGURES

Figure 1. Activity diagram of current system	2
Figure 2. Pie chart for Android version distribution in Google Play Store.....	4
Figure 3. Activity diagram for Go Guro	6
Figure 4. Use case diagram for Go Guro	7
Figure 5. Main activity and map activity screenshots of Go Guro	8
Figure 6. Project timetable from week 1 to week 2.....	8
Figure 7. Project timetable from week 3 to week 4.....	9
Figure 8. Project timetable from week 5 to week 6.....	9
Figure 9. Project timetable from week 7 to week 8.....	9
Figure 10. Project timetable from week 9 to week 10.....	9
Figure 11. Google Form questionnaire with first two statements	11
Figure 12. Bar chart for statements one to five of Google Form survey.....	12
Figure 13. Bar chart for statements six to ten of Google Form survey	12
Figure 14. System Usability System Calculations spreadsheet by Satori Interactive with values from respondents	15
Figure 15. Login/register form	17
Figure 16. Main activity screenshot of Go Guro	31
Figure 17. Pointing to "REGISTER" button.....	32
Figure 18. Main activity screenshot of Go Guro	32
Figure 19. Pointing to "LOGIN" button.....	33
Figure 20. Pointing to "FORGOT PASSWORD?" button	33
Figure 21. Reset password email from Firebase	34
Figure 22. Reset password form	34
Figure 23. Reset password form filled in	34
Figure 24. Password changed notification.....	35
Figure 25. Pointing to "SETTINGS" button	35
Figure 26. Pointing to profile picture placeholder	36
Figure 27. Pointing to "CONFIRM" button	36
Figure 28. Pointing to "SETTINGS" button.....	37
Figure 29. Pointing to "RESET PASSWORD" button	37
Figure 30. Reset password email from Firebase	38
Figure 31. Reset password form	38
Figure 32. Reset password form filled in	38
Figure 33. Password changed notification.....	39
Figure 34. Pointing to "OFF" toggle button	39
Figure 35. Pointing to "ACCEPT" and "DECLINE" buttons	40
Figure 36. Pointing to "START SESSION" button	40
Figure 37. Pointing to "END SESSION" button.....	41
Figure 38. Pointing to "What do you need help with?" textbox.....	41
Figure 39. Pointing to "CALL FOR HELP" button	42
Figure 40. Pointing to "SEND MESSAGE" button	42
Figure 41. Pointing to "GETTING YOUR TUTOR" button.....	43
Figure 42. Pointing to "HISTORY" button	43

Figure 43. Pointing to single history item..... 44

Figure 44. Single history page 44

Figure 45. Pointing to star rating bar 45

Figure 46. Pointing to “LOGOUT” button..... 45

Figure 47. Use case diagram for Go Guro 47

Chapter I

THE PROBLEM DOMAIN

A. Statement of the Problem

Students who have difficulties in a subject might need help from a nearby teacher or fellow student. Finding a nearby tutor is a problem and involves plenty of walking around the campus and asking people. This is difficult especially when class hours are over, and your classmates have already gone home.

There is also the fear of rejection, one of our deepest human fears, that will discourage those who are looking for a tutor because of getting rejected in-person. Fear of rejection falls under the basic fear of separation— “the fear of abandonment, rejection, and loss of connectedness; of becoming a non-person—not wanted, respected, or valued by anyone else.” This basic fear ranks second in the hierarchy of fear (Albrecht, 2012).

B. Background and Objectives of the Project

Because instances of irregular students (those who failed to follow the recommended course sequence due to failures or dropping courses; course shifters) are common in colleges, there is a need to develop a system that will help not only irregular students but also regular students in finding tutors without worrying about getting rejected personally and without needlessly going around the campus.

Therefore, the objectives of the project are:

- to reduce the time it takes to find a tutor,
- to reduce the number of footsteps it takes to find a tutor,
- and; to lessen the tutee's fear of rejection.

C. Significance and Scope of the Project

Go Guro solves the problem of fear of rejection by using anonymity. The names and faces of the tutors and tutee are not shown to each other until the tutor accepts the tutee's invitation. This, in effect, almost eliminates the tutee's fear of personal rejection. The application does not show to the tutee that tutors who have declined but only the tutor who accepted.

The application chooses tutors based on one parameter only: proximity to the tutee. This ensures that the closest possible tutor is found therefore reducing physical movements for the tutee.

When fully implemented, this project can help students in their academics and can also boost their interpersonal skills due to student-student or student-teacher interactions.

Go Guro is limited to only finding tutors. Monitoring of the study session is out of the project's scope.

D. Documentation of Existence and Seriousness of the Problem

The current system of finding tutors is to move around the campus and ask other students or professors if they are willing to help them in a topic. The problem with this is that it can be tiresome for the tutee to walk around the campus looking for tutors. Those that qualify to be a tutor are often invisible to tutees looking for them because they do not know they are able to help them.

The human fear of rejection is also a problem because it discourages tutees to use this system due to fear of being rejected in-person. This fear extends to an alternative system that uses social media or messaging sites like Facebook, Twitter, Messenger, or Viber to privately contact a student or teacher and ask for help. There is a possibility that the tutee can still be rejected with this alternative system, albeit not face-to-face.

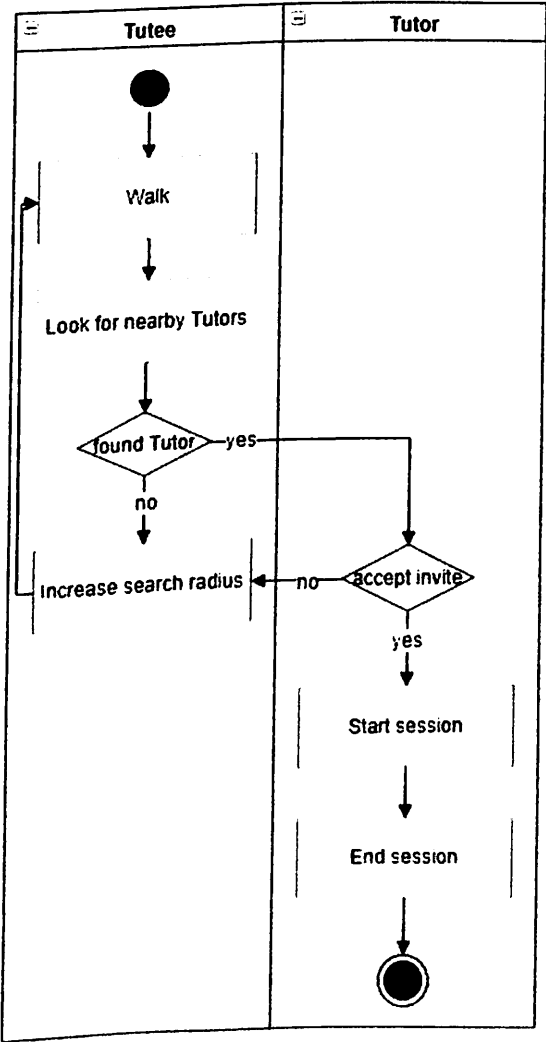


Figure 1. Activity diagram of current system

Chapter II

REVIEW OF EXISTING ALTERNATIVES

There are two systems available for those looking for tutors. One system is purely manual and the other is a hybrid between manual and computerized.

Manual system. The manual system involves walking around the campus to look for tutors and asking them in-person. This can be tiresome for the tutee and since the invitation will happen face-to-face, they might be discouraged to ask due to fear of rejection.

Hybrid system. This system uses social networks like Facebook and Twitter; instant messaging applications like Messenger, Viber, and WhatsApp; and SMS messaging. These applications are used to facilitate communication between tutee and tutor, but the tutee still has to ask the tutor. Even though the tutee and the tutor can be physically apart, the fear of rejection still influences the tutee. "Most people would probably expect that being ignored or rejected via a remote source like the Internet would not hurt as much as being rejected in person. Yet, our studies show that people may experience similar psychological reactions to online exclusion as they do with face-to-face exclusion (Smyth, 2012)."

The manual system has two problems in that it can be tiring for the tutee and the fear of rejection is apparent. The hybrid system solves the first problem of the manual system by using remote communication techniques, but the fear of rejection is still a problem with the hybrid system.

The proposed system was developed to be completely computerized through a mobile application. Like the hybrid system, remote messaging technologies is used but the tutee does not need to message the tutor personally; the application invites nearby tutors who also has the application with a pre-set message that tells them what the tutee needs help in. The names and faces of both tutee and tutor is not be shown to each of them; this approach lessens the fear of rejection by not attaching a person's name or face in an invitation.

Chapter III

APPROACH TO BE TAKEN IN THIS SUBJECT

A. Theoretical Framework

The system is an Android application targeted for mobile devices using at least Android 4.0.3 "Ice Cream Sandwich." This makes the system portable and usable in any location provided that the tutee has mobile data and location services enabled.

B. Rationale for the Framework

There are more Android devices in 2016 than any other platform (Gartner, 2017). This means that the system is compatible with more users. More users mean more adoption of the system.

By targeting Android 4.0.3 "Ice Cream Sandwich" as the minimum Android version, the system is guaranteed to work with phones from as early as October 2011. In addition, the system also targets Android 8.0 "Oreo" as the maximum. Using Google Play statistics gathered from February 5, 2018, Android 4.0.3 "Ice Cream Sandwich" constitutes to around 0.4% of visitors. Adding Google Play distribution percentages from Android 4.0.3 "Ice Cream Sandwich" to Android 8.0 "Oreo," we get around 99.3% of visitors with devices capable of using the application.

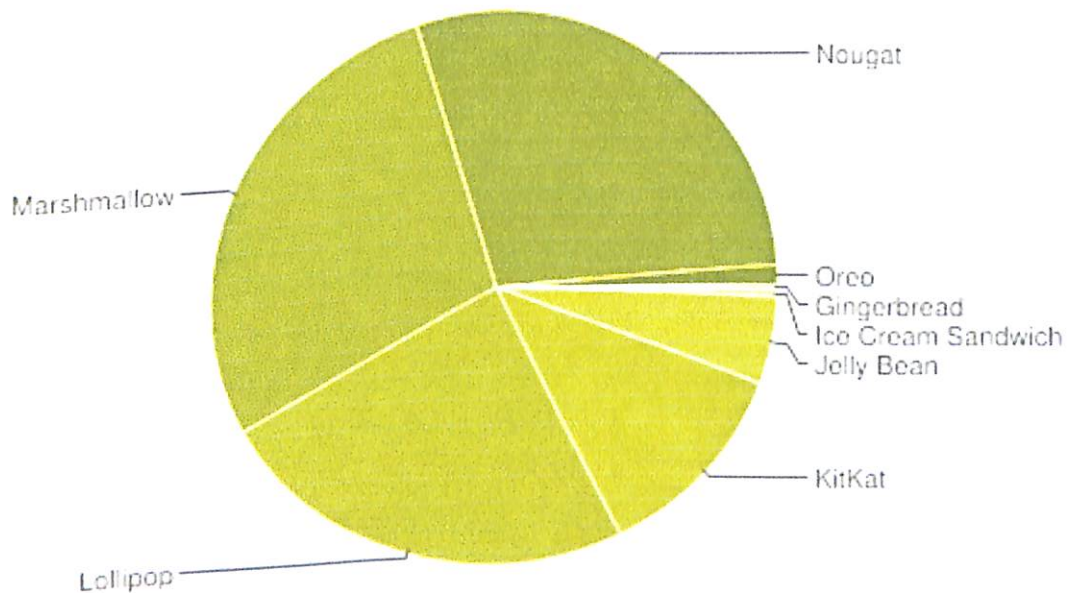


Figure 2. Pie chart for Android version distribution in Google Play Store

C. Technologies You Plan to Consider to Use

Android Studio 3.0.1 was used to build the application. This is the official IDE for Android. The system used APIs such as Google Maps API, Firebase, GeoFire, and Glide. Google Maps API, together with GeoFire, was used to display a map and show the location of nearby tutors. Firebase was used as a backend for the application that will handle tasks such as database management, authentication, etc. Glide was used for user profile photo management.

Chapter IV

PROJECT PLAN

A. Concept

Students will have a mobile application that can help them find tutors easily and lessen the fear of rejection by anonymously sending the invites for the study session. The application notifies tutors if they will accept or reject the invitation before meeting the tutee in-person. As mentioned in previous sections, what happens during study session is out of this project's scope.

Only if the invitation is accepted will the tutor and the tutee meet up personally. The activity diagram below lists the steps taken by the application in finding tutors.

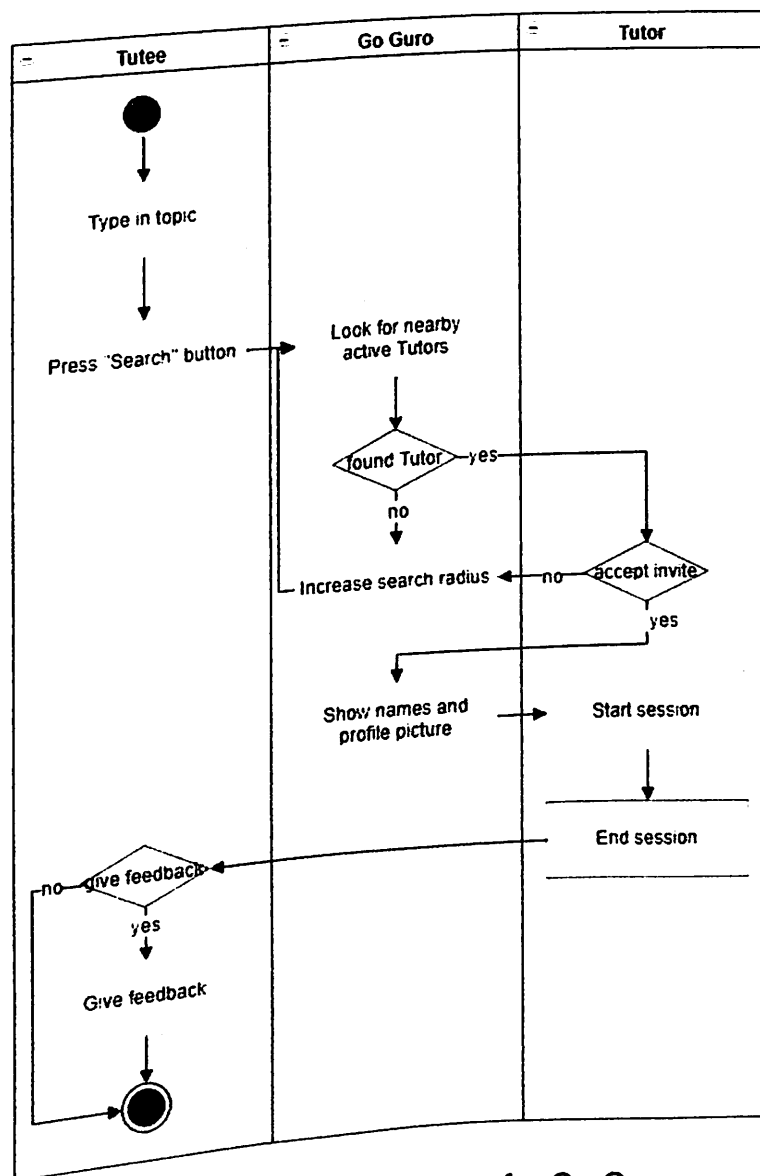


Figure 3. Activity diagram for Go Guro

The use case diagram below represents what can be done in the system. It's worth noting that the tutor can also be a tutee and therefore, can do use cases of the tutee.

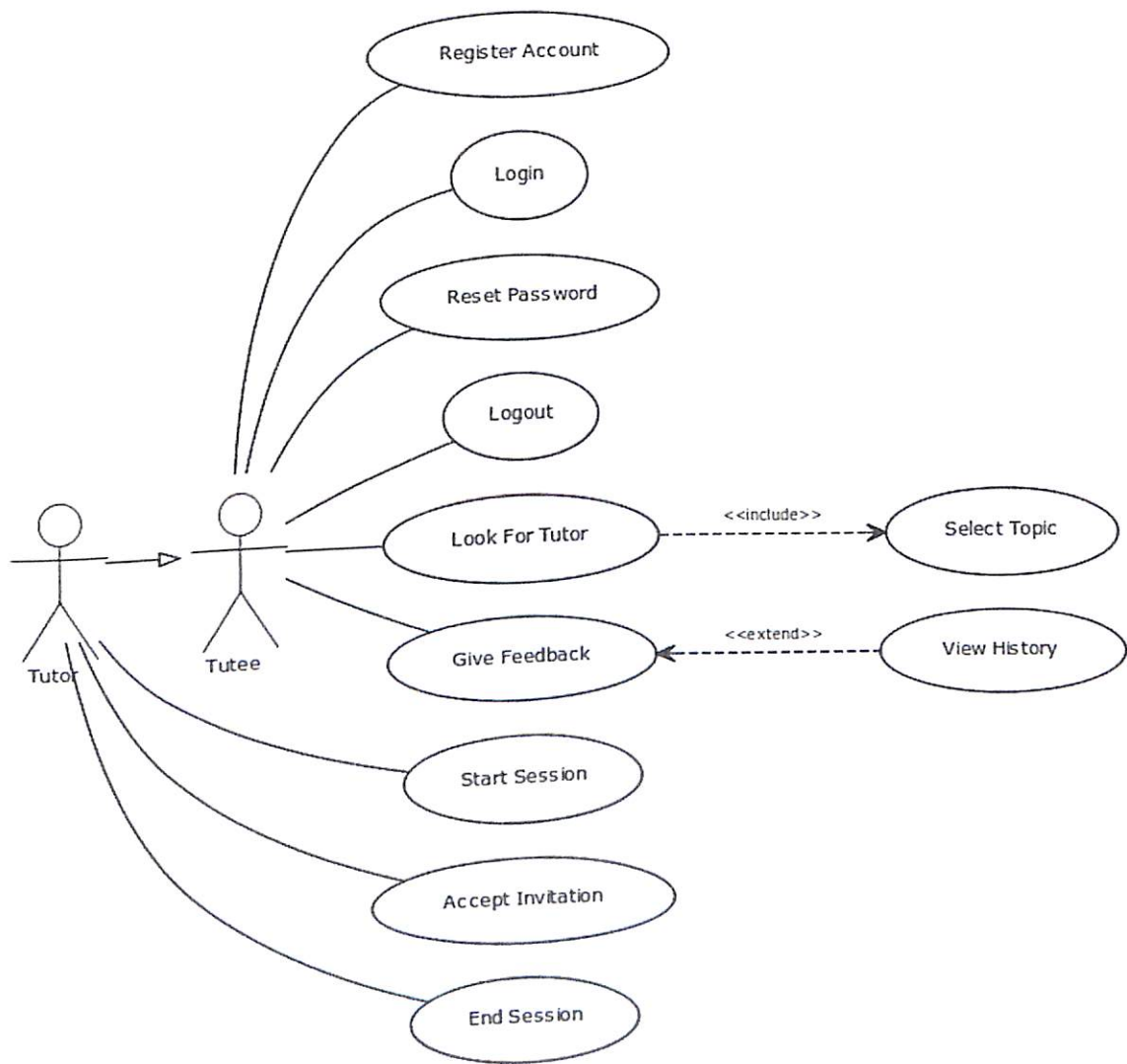


Figure 4. Use case diagram for Go Guro

Anonymity was done by not showing the name and face of the tutor to the tutee. The tutor must accept the invitation before they see the name and face of each other.

The interface was designed according to Google's Material Design guidelines. This can help in making the system user-friendly and therefore, more enjoyable and enticing to use. See the screenshots of the main activity and map activity of the application below:

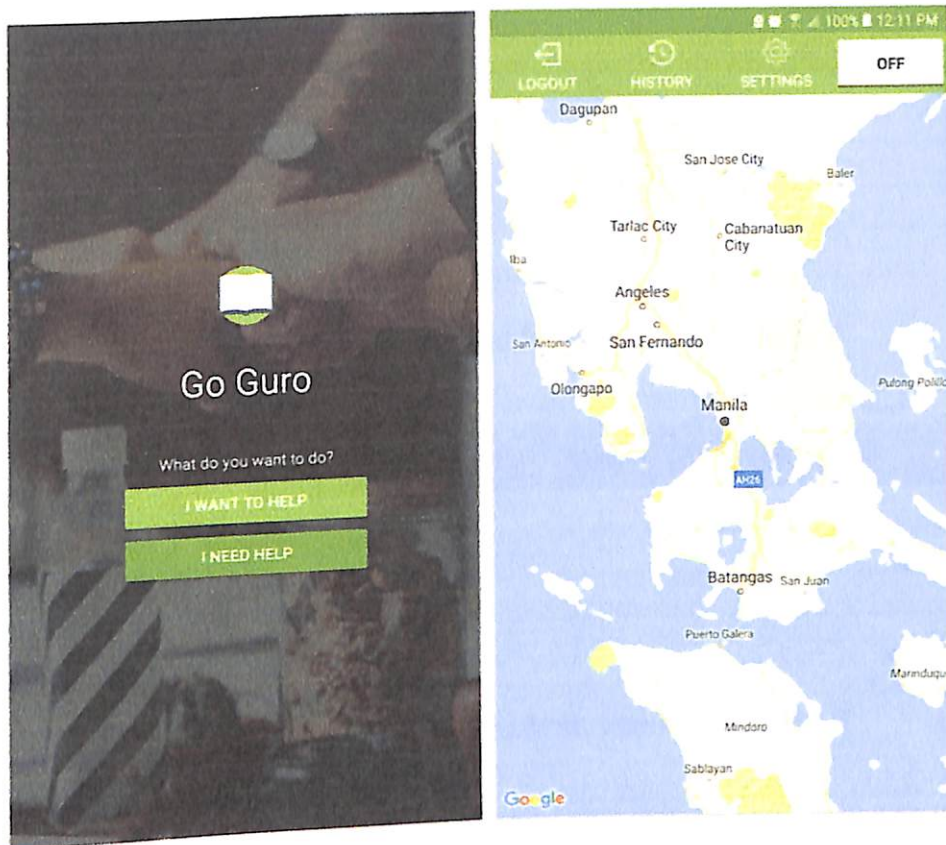


Figure 5. Main activity and map activity screenshots of Go Guro

B. Methods

Project development took 69 days or ten weeks. Programming the prototype alone took most of the project timetable, spanning 56 days or eight weeks. This was mainly due to having no prior experience with Android development. Video tutorials were extensively used to aid in learning Android programming. Because of this, prototype development took longer than expected.

Task	Start Date	End Date	Week 1								Week 2							
			2/5/18	2/6/18	2/7/18	2/8/18	2/9/18	2/10/18	2/11/18	2/12/18	2/13/18	2/14/18	2/15/18	2/16/18	2/17/18	2/18/18		
Develop prototype	05-Feb-18	01-Apr-18																
Demo prototype to end-users	02-Apr-18	02-Apr-18																
Revise prototype	03-Apr-18	06-Apr-18																
Create user manual	07-Apr-18	08-Apr-18																
Revise systems proposal	09-Apr-18	14-Apr-18																

Figure 6. Project timetable from week 1 to week 2

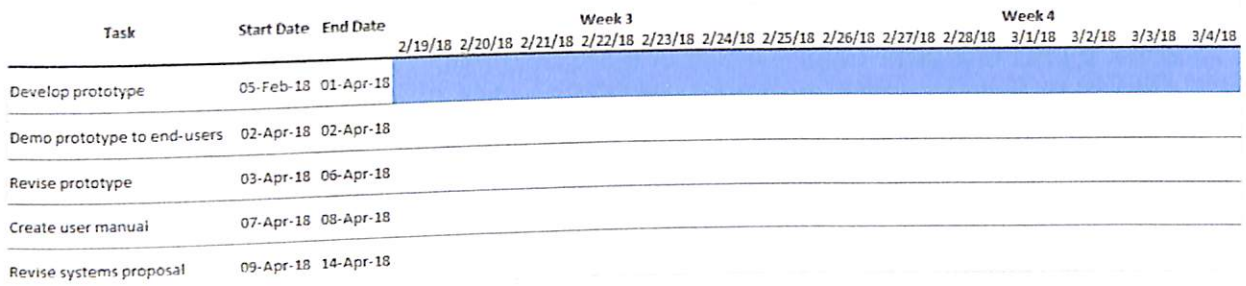


Figure 7. Project timetable from week 3 to week 4

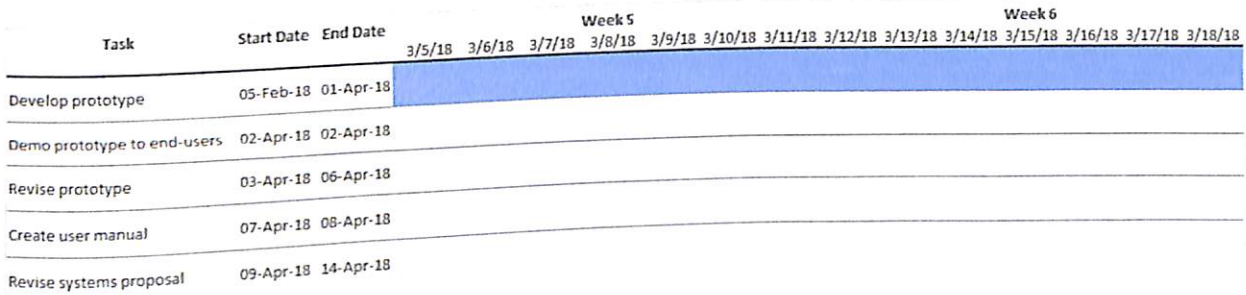


Figure 8. Project timetable from week 5 to week 6

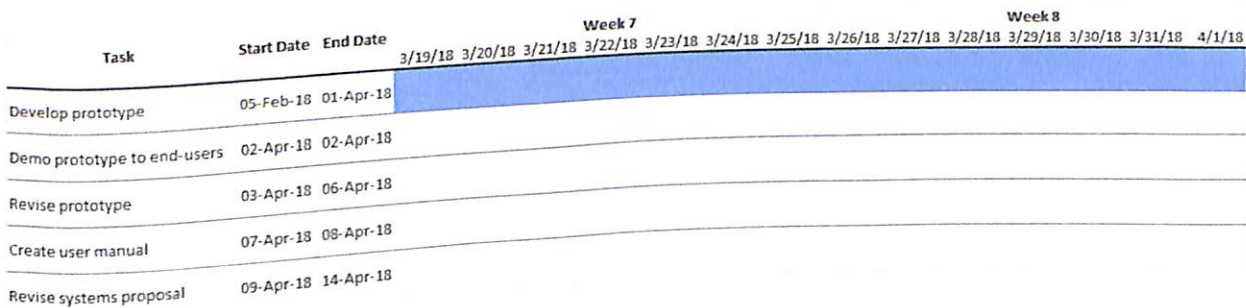


Figure 9. Project timetable from week 7 to week 8

The prototype was complete on April 1. However, there were only thirteen days left in the timetable. Hence, testing was done in one day only. Four days were allotted to prototype revision due to bugs found during the demo. Bugs were found because it was the first time the application was used outside of the development environment, where mobile data was used instead of WiFi.

User manual was created in two days. After that, the manuscript was revised on the last week up to April 14 submission date.

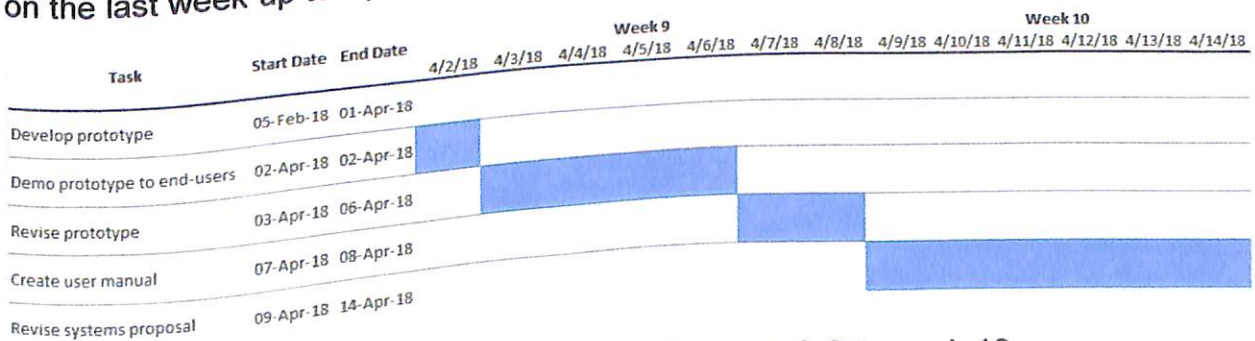


Figure 10. Project timetable from week 9 to week 10

C. Plan for User Testing and Project Assessment

Testing was done on AMA Computer College – Las Piñas and De La Salle – College of Saint Benilde. 9:00 AM to 12:00 NN time period was dedicated to AMA Computer College – Las Piñas, and 3:00 PM to 6:00 PM time period was dedicated to De La Salle – College of Saint Benilde. 12:00 NN to 03:00 PM was accounted for lunch break and travel time from Las Piñas to Taft Avenue, Manila.

A total of 21 persons took part in testing. First, they were briefed with the goal and methodology of the test. They were given at most thirty minutes to use Go Guro on a control device. They were instructed to use the app as a tutee while the author acted as the tutor, and vice-versa. Once they were ready to answer the questionnaire, they were redirected to an online survey using Google Forms on the same control device.

The questionnaire was made according to the SUS - a quick and dirty usability scale (Brooke, 1986), freely provided by Usability.gov. The System Usability Scale was created by John Brooke in 1986. It is a Likert scale consisting of ten items. These items are the following:

1. I think that I would like to use this mobile app frequently.
2. I found this mobile app unnecessarily complex.
3. I thought this mobile app was easy to use.
4. I think I would need assistance to be able to use this mobile app.
5. I found the various functions in this mobile app were well integrated.
6. I thought there was too much inconsistency in this mobile app.
7. I would imagine that most people would learn to use this mobile app very quickly.
8. I found this mobile app very cumbersome to use.
9. I felt very confident using this mobile app.
10. I needed to learn a lot of things before I could get going with this mobile app.

In addition, there was a textbox on the end of the system usability scale where respondents can suggest improvements to the app. It was titled: "How would you improve this mobile app?"

Go Guro Usability Evaluation

Go Guro is an Android application that aims to provide students an easy way to call for a tutor. Whether you need help on a specific topic, a project, or for a peer review of your thesis, Go Guro is your app

This survey focuses on general ease of use of Go Guro. This can be accomplished in less than 5 minutes

Please submit feedback regarding the general usability of Go Guro using the System Usability Scale below

© Digital Equipment Corporation, 1986.

* Required

Email address *

System Usability Scale *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I think that I would like to use this mobile app frequently.	☐	☐	☐	☐	☐
I found this mobile app unnecessarily complex.	☐	☐	☐	☐	☐

Figure 11. Google Form questionnaire with first two statements

The questionnaire focused on the overall ease of use of Go Guro. The survey was not hard to set up and administer, but scoring it proved to be difficult. Because of this, a spreadsheet that automatically calculates the SUS score was used. This was provided by Satori Interactive.

Chapter V

RESULTS AND DISCUSSION

A. System Usability Scale Results

The figures below show bar charts. In the X-axis are the statements and in the Y-axis are the number of respondents. These charts were provided by Google Forms.

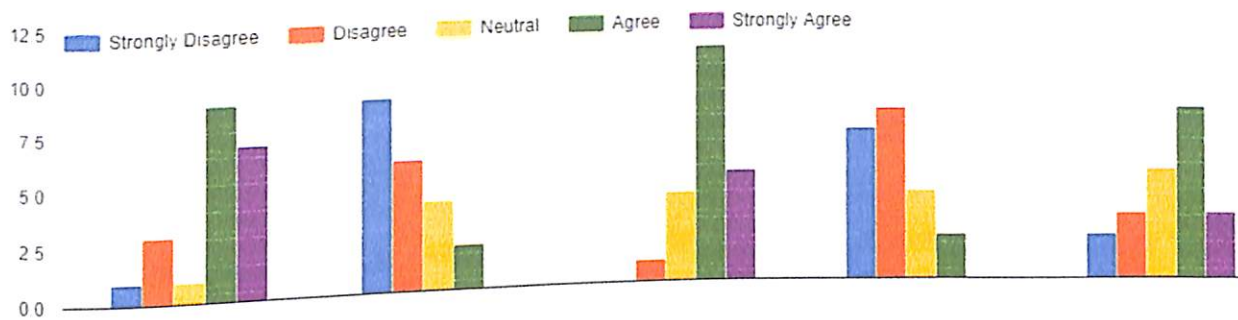


Figure 12. Bar chart for statements one to five of Google Form survey

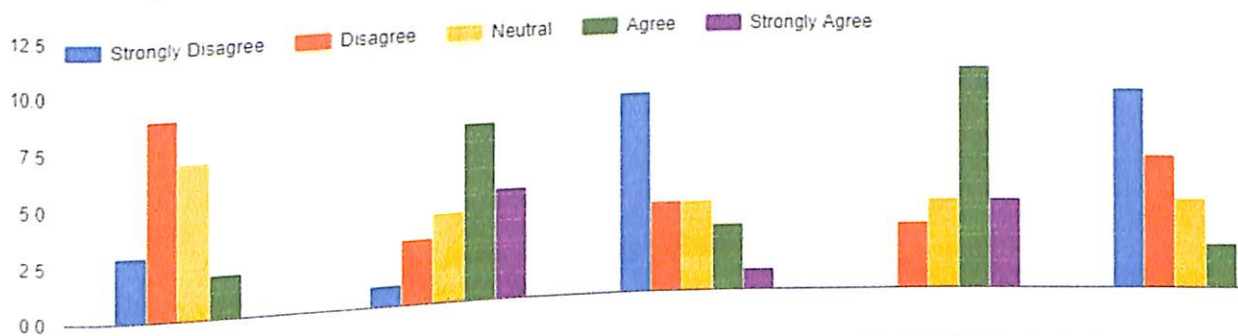


Figure 13. Bar chart for statements six to ten of Google Form survey

I think that I would like to use this mobile app frequently. Majority of respondents answered "agree" and "strongly agree." However, there were three respondents who answered "disagree," one who answered "neutral," and one who answered "strongly disagree."

I found this mobile app unnecessarily complex. Nine respondents answered "strongly disagree" and six with "disagree." Four answered "neutral" and two that answered "agree." Still, majority are in disagreement with this statement.

I thought this mobile app was easy to use. Majority of respondents agreed, having eleven respondents answered "agree" and five answered "strongly agree."

I think I would need assistance to be able to use this mobile app. Majority of respondents disagreed, with eight respondents answered "disagree" and seven answered "strongly disagree."

I found the various functions in this mobile app were well integrated. Three answered "strongly agree" and eight answered "agree." Three answered "disagree" and two answered "strongly disagree." Nonetheless there are five who answered "neutral."

I thought there was too much inconsistency in this mobile app. Many respondents disagree with this statement, but seven respondents were neutral.

I would imagine that most people would learn to use this mobile app very quickly. Majority agrees with this statement, with five "strongly agree" and eight "agree" answers. Four answered "neutral," three answered "disagree," and one answered "strongly disagree."

I found this mobile app very cumbersome to use. Nine respondents answered "strongly disagree," four answered "disagree" and "neutral," three answered "agree," and one answered "strongly agree."

I felt very confident using this mobile app. A lot of respondents agree with this statement. Ten answered "agree" and four answered "strongly agree." Four answered "neutral" and three answered "disagree."

I needed to learn a lot of things before I could get going with this mobile app. Most are in disagreement with this statement. Nine respondents answered "strongly disagree" and six answered "disagree." There were also four who answered "neutral" and two who answered "agree."

B. System Usability Scale Results Interpretation

The table below shows the corresponding point each option on the statements in the questionnaire:

Option	Value
Strongly Disagree	1
Disagree	2
Neutral	3
Agree	4
Strongly Agree	5

Table 1. Point value of each option in System Usability Scale

Using this, we can compute the scores for each statement. For odd numbered statements (1, 3, 5, 7, 9), the score is the value minus one. For even numbered statements (2, 4, 6, 8, 10), the score is five minus the value. The scores are within the range 0 to 4. Once all statements' scores have been computed, they are added together then multiplied by 2.5. This provided the overall system usability score for

one respondent. The table below shows an example computation for one respondent.

Number	Strongly Disagree			Strongly Agree		Value	Calculation	Score
1	☐	☐	☐	☒	☐	4	4-1	3
2	☒	☐	☐	☐	☐	1	5-1	4
3	☐	☐	☐	☐	☒	5	5-1	4
4	☒	☐	☐	☐	☐	1	5-1	4
5	☐	☒	☐	☐	☐	2	2-1	1
6	☒	☐	☐	☐	☐	1	5-1	4
7	☐	☐	☐	☒	☐	4	4-1	3
8	☒	☐	☐	☐	☐	1	5-1	4
9	☐	☐	☐	☒	☐	4	4-1	3
10	☐	☐	☒	☐	☐	3	5-3	2
							Sum	32
							x 2.5	80

Table 2. Score computation for one respondent

Once the scores for all respondents have been computed, the final score for all respondents are averaged and the overall score is produced.

The computation was done using Satori Interactive’s System Usability System Calculations spreadsheet that has the computations built-in. The survey information was extracted from Google Forms and put into the spreadsheet. The figure below shows the overall system usability score.

System Usability System Calculations



Simply enter the SUS scale position for each statement and participant in the corresponding fields. This spreadsheet will automatically calculate the system usability score for each participant.

**NOTE: If you have more than 5 participants, add them beneath the 5th one here, copy & paste the equation for the SU score, and update the average calculation to account for the proper number of participants. If you have questions about a particular situation, feel free to contact us as info@satoriinteractive.com.*

Participant	Statement 1	Statement 2	Statement 3	Statement 4	Statement 5	Statement 6	Statement 7	Statement 8	Statement 9	Statement 10	SU Score
jomar diamse{	4	1	5	1	2	1	4	1	4	3	80.0
chuckie0029@	2	3	4	2	4	2	3	1	4	1	70.0
sherlyn a kanr	5	1	5	1	5	1	5	1	5	1	100.0
pacardo james	4	1	4	1	3	2	3	1	5	1	82.5
jannahlyka_pa	5	2	5	2	2	2	5	1	5	1	85.0
angelina librar	4	2	4	2	3	2	4	2	3	2	70.0
jorvi jangayo@	2	1	4	1	4	2	4	1	4	3	75.0
bernel0219@g	5	1	4	2	3	2	4	1	4	1	82.5
jsc.flores@gm	5	1	4	2	4	3	5	1	4	2	82.5
cpido@gmail	5	2	4	1	4	2	5	2	4	1	85.0
angelica nielo	3	2	4	2	3	3	2	2	3	3	57.5
ginatan31@ya	4	2	4	2	4	3	4	3	4	2	70
lester_bilon@	5	1	5	1	5	1	5	1	5	1	100
gabfab42@gm	4	2	4	1	5	2	3	2	3	1	77.5
brian_paglinav	4	4	3	2	2	3	4	4	4	2	55
janelly garchi@	4	3	4	3	4	3	4	3	4	3	62.5
gracepamaus'	4	3	2	3	4	3	2	3	4	1	60
richmond yu_2	5	3	5	4	1	2	4	5	3	2	52.5
randombadger	2	1	3	3	1	4	1	4	2	4	30
alambirakarol@	1	1	3	3	4	4	3	4	2	4	45
gilbertgulliverc	4	3	3	4	3	3	2	3	2	2	45
AVERAGE											69.88

Figure 14. System Usability System Calculations spreadsheet by Satori Interactive with values from respondents

The calculation yielded a score of 69.88. This is an exceptional score because any score above 68 is above average (Frieze, 2015). By scoring 69.88 means that Go Guro is usable to its end-users.

Chapter VI

CONCLUSIONS

With Go Guro scoring 69.88 in the system usability scale, denoting that it is highly usable, the application proved to be useful to its end-users. During testing, while testers were using the app, they indicated that it would be very practical in their school. They agreed that by not showing the name and faces of invited tutors, the tutees are less affected by rejection fears. They also agreed that movement around the campus are reduced when using the app since they tutor and tutee's location are placed on a map. Hence, the tutor and tutee know exactly where to go. There is no need to roam around needlessly, trying to ask strangers to help them and hoping not to get turned down. They also noted that the application would be extremely useful for shy students who are not confident enough to ask for other people's help. By using the application, these students can get help without asking people face-to-face.

They added that they would love the application to be on Google Play for free for everyone, so that students and teachers from other schools can use it too, not just within their college. Certainly, publishing the application on Google Play is possible.

With testing done and end-users accepting the usefulness of the application, it can be concluded that Go Guro has achieved its goals of providing a means for students to look for help anywhere, anytime, and without fear of being rejected.

Nonetheless, there is still room for improvement as some students gave suggestions for the application. These are elaborated in the succeeding section.

Chapter VII

RECOMMENDATIONS

On the survey, a textbox was provided to the respondents where they can give feedback and ideas for improvement of Go Guro. During testing, there were nine respondents who gave their suggestions using this textbox. Collectively, the recommendations were to:

- make the application name more memorable,
- make registration and login form separate,
- add helpful warning or error messages and prompts,
- add animation,
- make font size bigger in some pages,
- and; make iOS version.

Of these suggestions, three respondents suggested to make the registration and login separate forms. This was because the app was at the time using a single form that contains email and password fields. These fields were shared with login and registration functions. You can see form in the screenshot below:

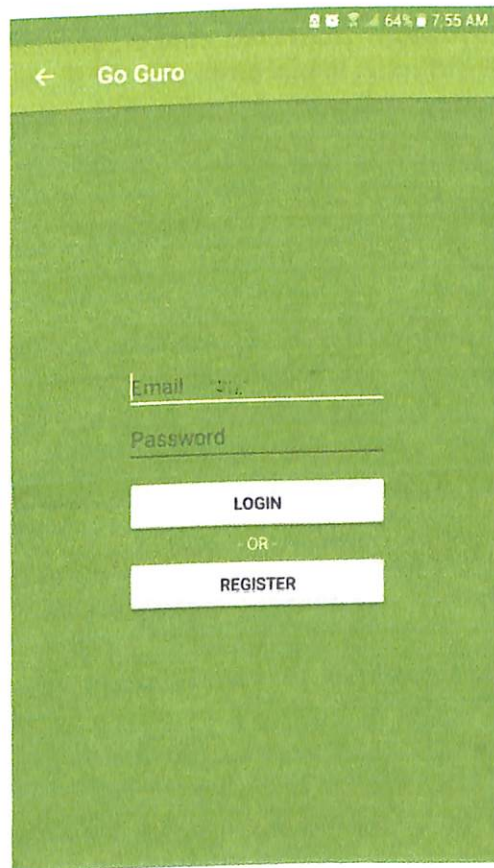
A screenshot of a mobile application interface for 'Go Guro'. The background is a solid green color. At the top, there is a white header bar with a back arrow on the left and the text 'Go Guro' in the center. Below the header, there are two white input fields: the first is labeled 'Email' and the second is labeled 'Password'. Below these fields, there is a white button labeled 'LOGIN'. Underneath the 'LOGIN' button, there is a small white box containing the text 'OR'. Below that, there is another white button labeled 'REGISTER'. The status bar at the very top of the screen shows icons for signal, Wi-Fi, and battery, along with the text '64%' and '7:55 AM'.

Figure 15. Login/register form

The author agrees that this is a sound suggestion and will certainly be taken into consideration with the next update of Go Guro.

A. Database Remodel

In addition, at the time of writing this, the tutors and tutees are separate entities in the database. The list below shows the database structure:

- History
- Topics
- Users
 - Tutors
 - 50G54XQE21gGdLA1XRfvvrQr193
 - Name: "Jeano Ermitano"
 - Phone: "09165467812"
 - profileImageUrl: "https://firebasestorage.googleapis.co..."
 - Tutees
 - 566nCuxXu6TjEoXvAtxWiHOaWPw1
 - Name: "Karol Alambra"
 - Phone: "09224802156"
 - profileImageUrl: "https://firebasestorage.googleapis.co..."

This design effectively separates the roles. Moreover, users can still switch from tutor to tutee. However, if for example Karol Alambra switches from tutee to tutor, a record in the Tutors node is created using her user ID. The updated structure would be:

- History
- Topics
- Users
 - Tutors
 - 50G54XQE21gGdLA1XRfvvrQr193
 - Name: "Jeano Ermitano"
 - Phone: "09165467812"
 - profileImageUrl: https://firebasestorage.googleapis.co...
 - 566nCuxXu6TjEoXvAtxWiHOaWPw1
 - Tutees
 - 566nCuxXu6TjEoXvAtxWiHOaWPw1
 - Name: "Karol Alambra"
 - Phone: "09224802156"
 - profileImageUrl: "https://firebasestorage.googleapis.co..."

Currently, the newly created account on Tutors node does not carry over the account information from the Tutees node and vice-versa. Therefore, the user needs to supply the account information again. This was also noticed during the testing period by the respondents.

Furthermore, this design is not efficient because there is redundant information. The database structure needs to be remodeled such that redundant records are reduced, if not eliminated.

B. iOS Port

A couple of testers suggested to have Go Guro available on Apple devices as well. While this would mean more work for the author, as he has to development, maintain, and update for two platforms, this is a good suggestion. This however, would mean a completely different development environment. A Mac is required to run Xcode, the only IDE for iOS and Mac app development. An Apple developer license and an iOS device are not required to develop the prototype; a virtual iOS device would be sufficient. Nonetheless, they are needed when publishing to the App store. Aside from this, the suggestion is ideal especially when there are more developers aside from the author.

Chapter VIII

REFERENCES

- Gartner (2017). *Gartner Says Worldwide Sales of Smartphones Grew 7 Percent in the Fourth Quarter of 2016*. Egham, U.K.: Gartner. Available: <http://www.gartner.com/newsroom/id/3609817>. [Accessed 23 September 2017]
- Albrecht, Karl (2012). *The (Only) 5 Fears We All Share*. Online: PsychologyToday. Available: <https://www.psychologytoday.com/blog/brainsnacks/201203/the-only-5-fears-we-all-share>. [Accessed 01 October 2017]
- Smyth, Joshua; Filipkowski, Kelly (2012). *Being ignored online or in person, it's still exclusion*. University Park, Pennsylvania, USA: Pennsylvania University. Available: <http://news.psu.edu/story/150196/2012/04/03/research/being-ignored-online-or-person-its-still-exclusion>. [Accessed 02 October 2017]
- SimCoder (2017). *Make an Android App Like UBER - Part 1 – Introduction*. Online: Youtube. Available: <https://www.youtube.com/watch?v=Z3HdOTQ8YHI>. [Accessed 30 September 2017]
- Udacity. *Developing Android Apps by Google*. Online: Udacity. Available: <https://www.udacity.com/course/new-android-fundamentals--ud851>. [Accessed 01 October 2017]
- Usability.gov. *System Usability Scale*. Online: Usability.gov. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> [Accessed 2 April 2018]
- John Brooke (1986). *SUS - A quick and dirty usability scale*. Reading, U.K.: Redhatch Consulting Ltd. Available: <http://www.usabilitynet.org/trump/documents/Suschapt.doc> [Accessed 2 April 2018]
- Stephanie Frieze (2015). *System Usability Scale: A Quick Usability Scoring Solution*. Online: Satori Interactive. Available: <http://satoriinteractive.com/system-usability-scale-a-quick-usability-scoring-solution-2/> [Accessed 2 April 2018]
- Satori Interactive. *System Usability System Calculations*. Online: <http://satoriinteractive.com/wp-content/uploads/2018/03/Satori-SUS-Calculator-1.xls> [Accessed 2 April 2018]

Chapter VIV
APPENDICES

A. Deliverables and Milestones

The manuscript was the main deliverable. In addition, the software requirements specifications was also attached to it. The source code of the completed Android application was attached using a link to the online GitHub repository together with a CD. Results of user testing was included in the manuscript as well as interpretation of these results under Results and Discussions section. There were suggestion by respondents to improve the application and these were included in the manuscript under Recommendations section. To summarize, the following were delivered in IS295b:

- Manuscript
- Software requirements specifications
- Project source code
- User testing results

B. Budget

Use of Android Studio is free. Likewise, Firebase also has a free tier that comes with the essential tools needed by this proposal. Therefore, prototype development did not cost anything. The only capital cost, if the application would be turned over to another developer, would be for Internet connection and a computer. The computer need not be powerful, and the Internet subscription need not be fast.

Upon searching for a laptop that meets the recommended specification of Android Studio and emulator, for a reasonable price, on three online retailers (PC Express, DynaQuest PC, Asianic), the Lenovo 310-14ISK comes out as being the cheapest while having the recommended specs. DynaQuest PC offers the lowest price on the laptop at only Php 32000. Below are the specifications of the unit:

- Intel® Core™ i5-6200U
- 8GB DDR4 RAM
- 1TB hard disk drive
- Windows® 10 Home 64-bit
- NVIDIA® GeForce® 920MX with 2GB VRAM
- 14" screen with 1366 x 768 resolution

With regards to subscription to an Internet service, a pocket WiFi service is most suited for this proposal. Comparing pocket Wifi service providers, Smart's postpaid Plan 299 is the least expensive that also offers a reasonable monthly data allowance. Below are the details about this Internet plan:

- 3.5GB monthly data allocation
- No one time device fee
- Php 299.00 monthly payment
- Total initial cash out: Php 299.00
- 12 months lock-in period

When the application is ready to leave testing phase to production, a Developer license is required to distribute the application in the Google Play store. Google charges a \$25 one-time fee. In addition, when the application exceeds the free tier of Firebase, a higher tier is needed which would \$25 monthly. If, in the future, the load exceeds this higher tier, a custom tier will be needed and the pricing will be adjusted in accordance to the load of the system.

Assuming the project was to be developed completely from scratch without a computer and Internet access, the following tables represent the initial and recurring expenses on this project. US Dollar conversions are computed using the rate of \$1 = Php 51.52.

Item	Price (Php)
Lenovo 310-14ISK laptop computer	32000.00
Smart Bro Pocket WiFi Plan 299	299.00
Google Developer license	1287.88
Firebase Flame plan	1287.88
Total	34874.76

Table 3. Initial project expenses

Item	Price (Php)
Smart Bro Pocket WiFi Plan 299	299.00
Firebase Flame plan	1287.88
Total	1586.88

Table 4. Recurring project expenses per month

C. Qualifications

Educational background. The author holds a bachelor's degree in Information Systems with De La Salle-College of Saint Benilde. Programming languages taught to him are listed below with each bullet representing a course (thus one course can have many languages taught), as well as the number of units required for each:

- Nine units (three courses)
 - C#
 - Java
- Six units (two courses)
 - SQL
- Three units (one course)
 - Objective-C, Swift
 - Javascript, HTML, CSS
 - PHP
 - WebGL, Ruby, Ruby on Rails

Apart from application development, he was also taught extensively the analysis and design phase of the software development lifecycle (SDLC). Core IS subjects in college that aided in the creation of this proposal include:

- Computer Systems Analysis
- Computer Systems Design and Quality Processes
- IT Documentation Standards
- Project Management

Java web apps. Prior experience with Java while the author was in college made this proposal viable for him. While he doesn't develop programs in his current job as web designer, the author has learned the basics of Java SE and EE; Java Server Pages, servlets, Hibernate ORM, and Struts 2. Him, along with two classmates, developed a movie ticketing web application as our Java machine project that uses QR codes for tickets, Paypal for payments, and Facebook for logins.

iOS mobile apps. With regards to mobile application development, the author has also studied Objective-C and Swift languages in a basic level which he used to develop simple iOS programs in college.

Android and Firebase. With the Java and mobile app development experiences that the author has, nonetheless this was his first project using Android. Therefore, he expanded his knowledge in programming with this capstone project. This project helped the author learn the Android platform and the features it has. In addition to Android Studio, the project used Firebase which was also his first encounter.

D. Contributors / Collaborators

Since the author had no experience in Android application development, he had to learn its basics. Udacity offers a free online course on developing Android apps. In addition, a Youtube channel by the name of SimCoder provides videos on Android application development, more specifically a series of videos on how to create a ride-sharing application. These greatly helped him in developing this project.

E. Resources

Draw.io. *unavailable*. Online: Draw.io. Available: <https://www.draw.io/>. [Accessed 25 September 2017]

yUML. *Draw Use Case Diagram*. Online: yUML. Available: <https://yuml.me/diagram/nofunky/usecase/draw>. [Accessed 25 September 2017]

Firebase. *Firebase helps you build better mobile apps and grow your business*. Online: Google. Available: <https://firebase.google.com/>. [Accessed 30 September 2017]

Android Developers. *Getting Started with Android Studio*. Online: Android Developers. Available: <https://developer.android.com/studio/index.html>. [Accessed 30 September 2017]

Google Developers. *Google Maps APIs*. Online: Google. Available: <https://developers.google.com/maps/documentation/android-api/>. [Accessed 30 September 2017]

Android Developers. *Introduction to Android*. Online: Android Developers. Available: <https://developer.android.com/guide/index.html>. [Accessed 01 October 2017]

Android Developers. *Package Index*. Online: Android Developers. Available: <https://developer.android.com/reference/packages.html>. [Accessed 01 October 2017]

Android Developers. *Class Index*. Online: Android Developers. Available: <https://developer.android.com/reference/classes.html>. [Accessed 01 October 2017]

DynaQuest PC. *Lenovo (310-14ISK 80SL000CPH) i5-6200U 8GB 920Mx 1TB Win10 14in*. Online: DynaQuest PC. Available: <http://dynaquestpc.com/product/lenovo-310-14isk-80sl000cph-i5-6200u-8gb-920mx-1tb-win10-14in/>. [Accessed 07 October 2017]

Smart Communications Inc. *Pocket Wifi*. Online: Smart Communications Inc. Available: <https://smart.com.ph/bro/pocket-wifi-gadgets/#slide/9> [Accessed 07 October 2017]

Firebase. *GeoFire — Realtime location queries with Firebase*. Online: GitHub. Available: <https://github.com/firebase/geofire> [Accessed 24 February 2018]

Bumptech. *Glide*. Online: GitHub. Available: <https://github.com/bumptech/glide> [Accessed 25 February 2018]

Android Developers. *Android 4.0 Compatibility Definition*. Online: Android Developers. Available:

<https://source.android.com/compatibility/4.0/android-4.0-cdd.pdf>
[Accessed 10 April 2018]

Android Developers. *Use Java 8 Language Features*. Online: Android Developers. Available: <https://developer.android.com/studio/write/java8-support.html> [Accessed 10 April 2018]

Android Developers. *Dashboards*. Online: Android Developers. Available: <https://developer.android.com/about/dashboards/index.html> [Accessed 10 April 2018]

F. Complete Program Listing

This project is hosted on GitHub and includes all source codes. Use the link below to access:

- <https://github.com/jeanoskii/Go-Guro>

The application uses Firebase to provide database management, file hosting, and monitoring. User registration, authentication, and password resets are handled by Firebase as well. Below are the Firebase APIs used by the application:

- `com.google.firebase:firebase-core:11.8.0`
- `com.google.firebase:firebase-database:11.8.0`
- `com.google.firebase:firebase-auth:11.8.0`
- `com.google.firebase:firebase-storage:11.8.0`

GeoFire was used to provide realtime location queries, which are used to plot pins on the map. This library uses Firebase to store coordinates. Below is the GeoFire API used in the system

- `com.firebase:geofire-android:2.3.0`

Google Maps API was used to draw a map on both tutor and tutee screens. Below are the APIs used:

- `com.google.android.gms:play-services-maps:11.8.0`
- `com.google.android.gms:play-services-location:11.8.0`

Glide is a library that provides a framework for image loading on Android devices. This was used on the application to manage user profile images. Below is the APIs used:

- `com.github.bumptech.glide:glide:4.0.0`

G. Technical Reference

Final system specifications. The system was programmed to target Android 8.0 “Oreo” with Android API level 26, but the system also supports devices from as early as Android 4.0.3 “Ice Cream Sandwich” with Android API level 15. With this, we can determine the minimum hardware requirements using the official Android Compatibility Definition Document for Android 4.0.3:

- Must have screen sizes of at least 426 dp x 320 dp
- Must support any form of 802.11 (for WiFi capability)
- Must support GSM or CDMA (for mobile data capability)
- Must include GPS receiver (for location service capability)
- Must have at least 340MB of memory
- Must have at least 350MB of storage

Devices must have an additional 5MB to install the application.

The system was developed using Android Studio 3.0.1 IDE with the Java programming language. Android Studio 3.0.1 supports Java 7 as well a subset of Java 8 language features, with more features being added on future releases of Android Studio.

As for the server, Firebase was used to provide real-time database management, file hosting (for profile images), monitoring, and user management including authentication.

Maintenance plan for the software system. To compile the system from source code, Android Studio 3.0.1 must be installed. Once installed, the project source code can then be imported to Android Studio. This can then be built to an .apk file which can be installed on Android devices. Detailed instructions on installation of Android studio can be found with the links below:

- <https://developer.android.com/studio/install.html>
- <https://developer.android.com/studio/projects/create-project.html#importAProject>
- <https://developer.android.com/studio/run/index.html#reference>

The system relies heavily on Firebase real-time database and Google Maps and Locations APIs to draw and plot user locations on the map. To store and display user profile images, Glide was used. Because of this, developers of this system should pay attention to interoperability of these external systems when they decide to upgrade to a later version. More information on these dependencies can be found here:

- <https://firebase.google.com/docs/android/setup>
- <https://developers.google.com/maps/android/>
- <https://github.com/bumptech/glide>

H. User Manual

Before using the app, the user must enable location services as well as mobile data or Wi-fi for Internet access. In addition, they must permit the app to use the device's current location.

Register. The steps below outline how to register an account.

1. Open app
2. Select action



Figure 16. Main activity screenshot of Go Guro

3. Type in email address
4. Type in password
5. Press "REGISTER" button

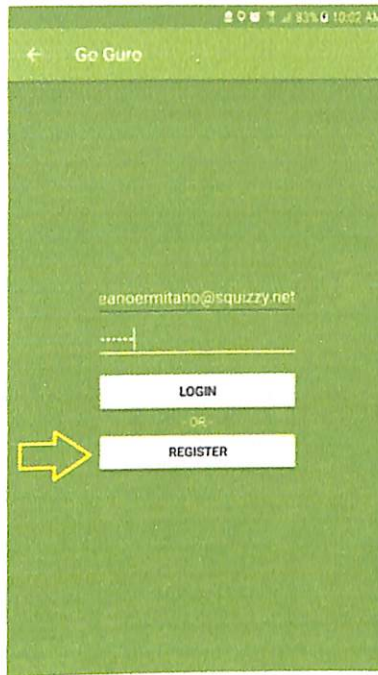


Figure 17. Pointing to "REGISTER" button

Login. The steps below outline how to login.

1. Open app
2. Select action

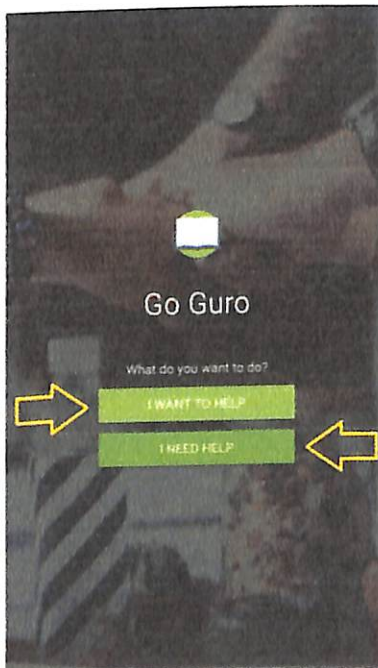


Figure 18. Main activity screenshot of Go Guro

3. Type in email address
4. Type in password
6. Press "LOGIN" button



Figure 19. Pointing to "LOGIN" button

Forgot Password. If a user forgets their password, the steps below outline how they can reset their password. The "FORGOT PASSWORD?" button will appear once there is a failed login attempt.

1. Press "FORGOT PASSWORD?" button

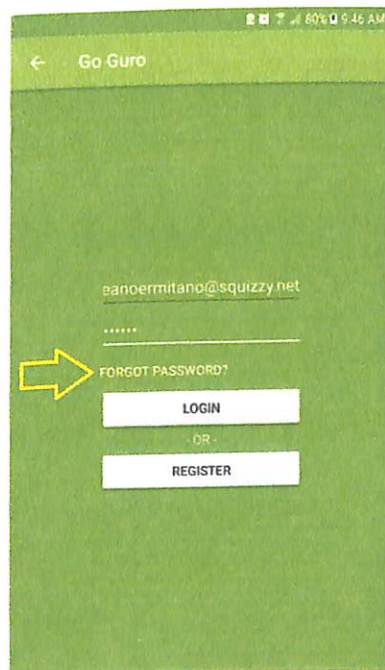


Figure 20. Pointing to "FORGOT PASSWORD?" button

2. Using an external system, login to email client using registered email address
3. Using the email client, open email for Reset Password from Google/Firebase

noreply@uber-abf8a.firebaseio.com

04/14/2018 03:43:35 AM

Reset your password for project-638648271363

Hello,

Follow this link to reset your project-638648271363 password for your jeanoermitano@squizzly.net account:

https://uber-abf8a.firebaseio.com/__/auth/action?mode=resetPassword&ocbCode=NeIF-0vxz7bKK05D6ZVv9mVfBH2X69_DOXvEtgiKrwUAAAFwdU9FA&apiKey=AlzaSyAcDPO5Lix8KjqE_dgvxyDbxEVnVveOsL0

If you didn't ask to reset your password, you can ignore this email

Thanks,

Your project-638648271363 team

Figure 21. Reset password email from Firebase

4. Using the email client, click link inside email
5. Once redirected, type in new password

Reset your password

for jeanoermitano@squizzly.net

New password



Figure 22. Reset password form

6. Press "SAVE" button

Reset your password

for jeanoermitano@squizzly.net

New password

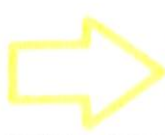


Figure 23. Reset password form filled in

Password changed

You can now sign in with your new password

Figure 24. Password changed notification

Complete details. After a user logs in or registers, they need to provide some information before using the app. The steps below outline how to complete these details.

1. Press “SETTINGS” button located at top-right corner of screen



Figure 25. Pointing to “SETTINGS” button

2. Press the placeholder image and select profile photo using Android image picker

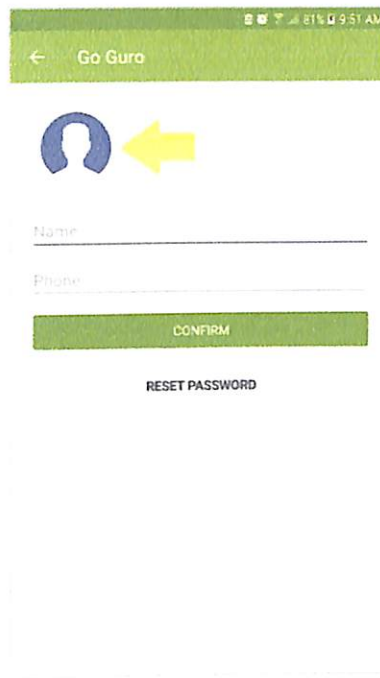


Figure 26. Pointing to profile picture placeholder

3. Type in name
4. Type in phone number
5. Press "CONFIRM" button



Figure 27. Pointing to "CONFIRM" button

Reset password. If a user wishes to change their password once logged in, the steps below should be followed.

1. Press "SETTINGS" button located at top-right corner of screen

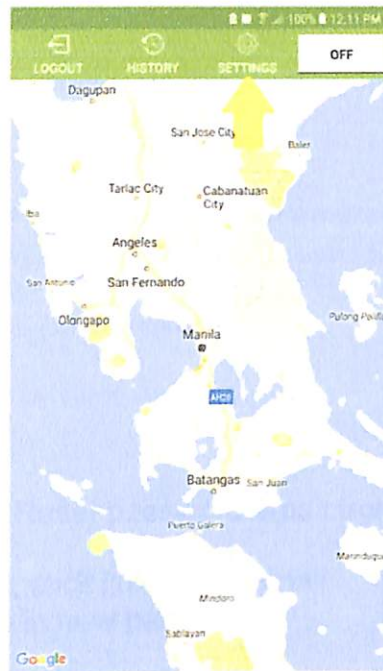


Figure 28. Pointing to “SETTINGS” button

2. Press “RESET PASSWORD” button



Figure 29. Pointing to “RESET PASSWORD” button

3. Using an external system, login to email client using registered email address
4. Using the email client, open email for Reset Password from Google/Firebase

noreply@uber-abf6a.firebaseio.com

04/14/2018 03:43:35 AM

Reset your password for project-638648271363

Hello,

Follow this link to reset your project-638648271363 password for your jeanoermitano@squizzly.net account:

https://uber-abf6a.firebaseio.com/_/auth/action?mode=resetPassword&ocbCode=NeIF-0vxz7bKK05D6ZVv9mVfBH2X69_DOXvEtgiKrwUAAAFwdU9FA&apiKey=AlzaSyAcDPO5Lix6KjqE_dgyxyDbxEVnVveOsL0

If you didn't ask to reset your password, you can ignore this email

Thanks,

Your project-638648271363 team

Figure 30. Reset password email from Firebase

5. Using the email client, click link inside email
6. Once redirected, type in new password

Reset your password

for jeanoermitano@squizzly.net

New password



Figure 31. Reset password form

7. Press "OK" button

Reset your password

for jeanoermitano@squizzly.net

New password



Figure 32. Reset password form filled in

Password changed

You can now sign in with your new password

Figure 33. Password changed notification

Teach Tutee. Once a tutor user finishes completing their details, they can now proceed to teach a tutee. This requires that a tutee is also looking for a tutor. The steps below outline how to do so.

1. Press "OFF" toggle button located at top-right corner of screen to turn it "ON"

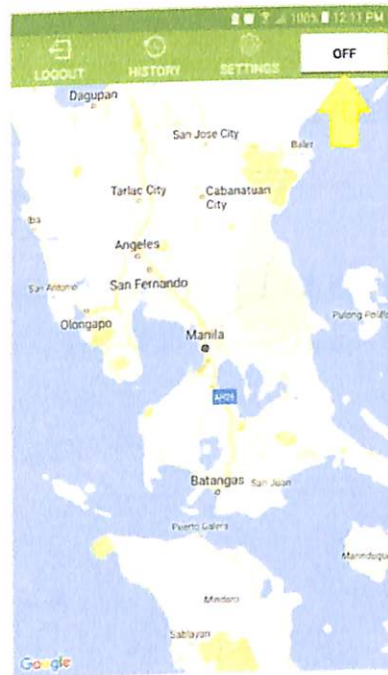


Figure 34. Pointing to "OFF" toggle button

2. Once a tutee is found, press "ACCEPT" button. Alternatively, tutor can press "DECLINE" button

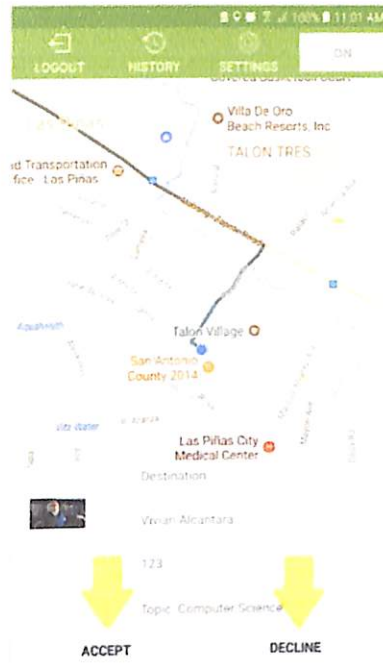


Figure 35. Pointing to “ACCEPT” and “DECLINE” buttons

3. Tutor and tutee converge locations
4. Press “START SESSION” button

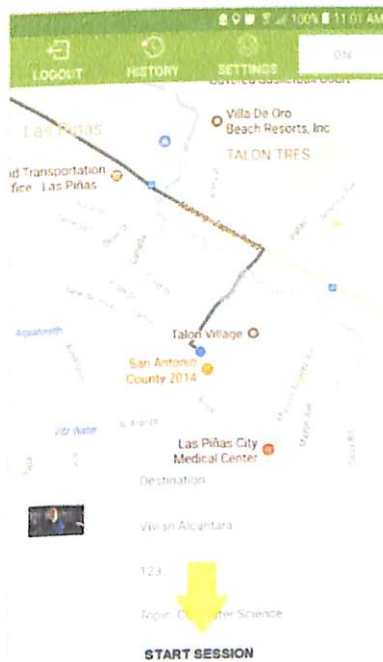


Figure 36. Pointing to “START SESSION” button

5. When study session has ended, press “END SESSION”

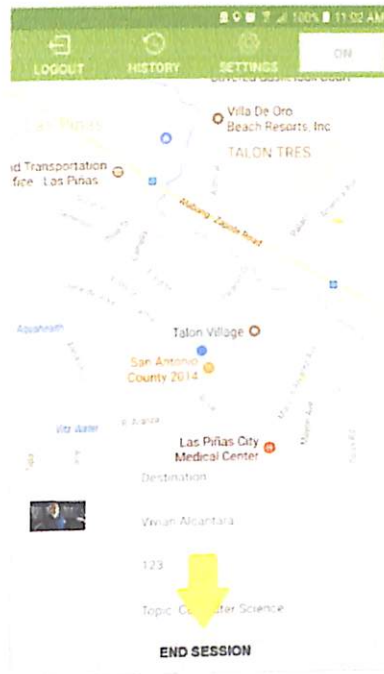


Figure 37. Pointing to “END SESSION” button

Search for Tutor. Once a tutee user finishes completing their details, they can now proceed to find a tutor. This requires an active tutor. The steps below outline how to do so.

1. Type in topic

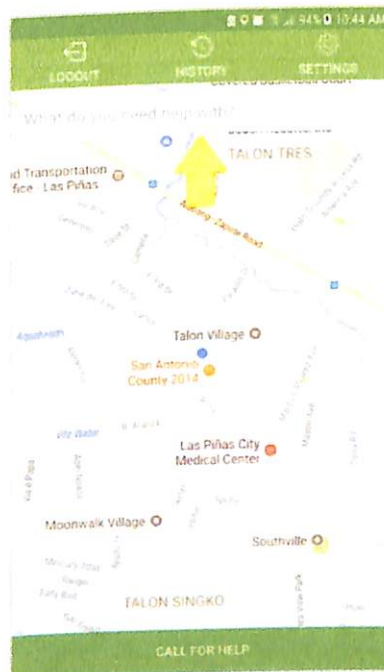


Figure 38. Pointing to “What do you need help with?” textbox

2. Press “CALL FOR HELP” button



Figure 39. Pointing to “CALL FOR HELP” button

3. Optionally, once a tutor is found, press “SEND MESSAGE” button to open default messaging app to send the tutor an SMS message



Figure 40. Pointing to “SEND MESSAGE” button

4. Tutor and tutee converge locations
5. Study session can now begin

Cancel Search. The steps below outline how a tutee can cancel searching for tutor. They must be actively searching for tutor first.

1. Press “GETTING YOUR TUTOR” button

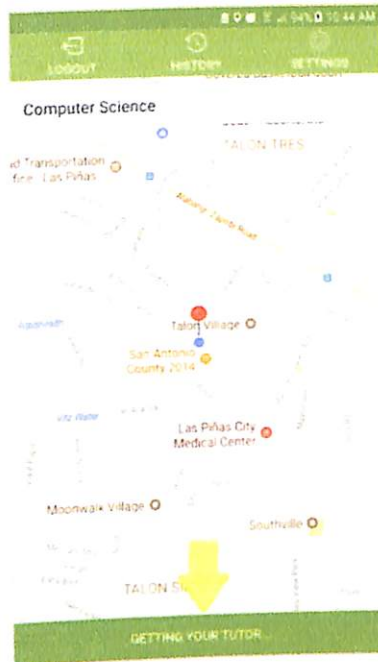


Figure 41. Pointing to “GETTING YOUR TUTOR” button

View previous study session. Once a study session is complete, users can view their past sessions. The steps below outline how to view previous study session.

1. Press “HISTORY” button on top-middle corner of screen

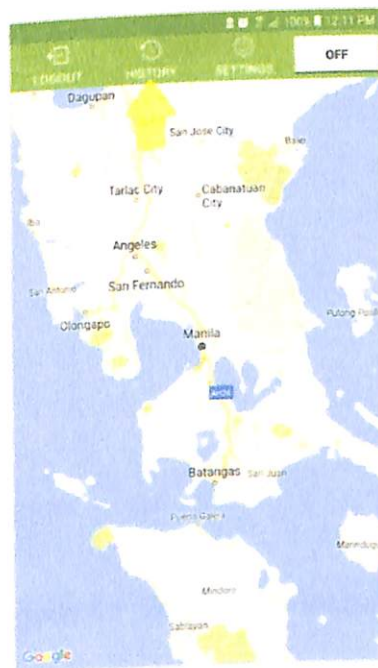


Figure 42. Pointing to “HISTORY” button

2. Press desired study session from list

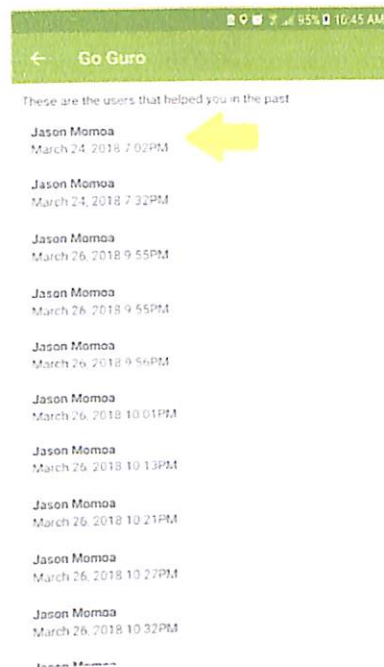


Figure 43. Pointing to single history item

3. View single study session details



Figure 44. Single history page

4. Optionally, the tutee can rate the tutor by pressing the desired star rating



Figure 45. Pointing to star rating bar

Logout. The step below outline how to log out of the app.

1. Press "LOGOUT" button on top-left corner of screen

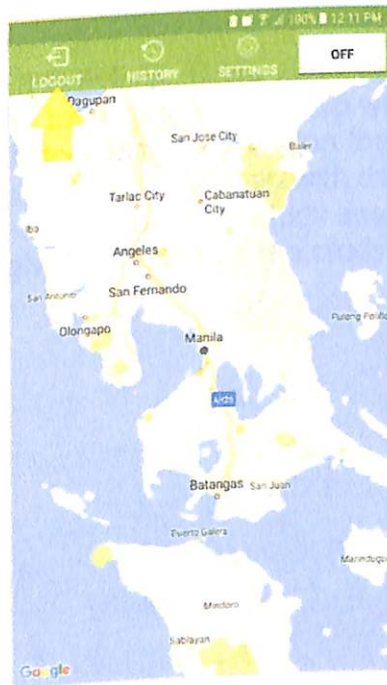


Figure 46. Pointing to "LOGOUT" button

I. Software Requirements Specification

Project: Go Guro

Internal Release Number: 0.0.1

Attached Worksheets: SRS > Use Case Suite
SRS > Feature Set

Use Case Suite.

Use Cases by Priority

- Essential
 - UC-00: Register Account
 - UC-01: Login
 - UC-02: Reset Password
 - UC-03: Logout
 - UC-04: Look for Tutor
 - UC-05: Start Session
 - UC-06: End Session
 - UC-07: Accept Invitation
- Expected
 - UC-08: Select Topic
- Optional
 - UC-09: View History
- Optional
 - UC-10: Give Feedback

Use Cases. There are two actors: Tutee and Tutor. The Tutee can look for a nearby Tutor after selecting the topics they need help with. The Tutor is a Tutee that the main Tutee will find. The Tutor gets a notification that informs him or her that the main Tutee is looking for a Tutor and needs help with the selected topics. When the Tutor accepts the invitation, they can start the study session. When the study session ends, the Tutee can give feedback on the study session.

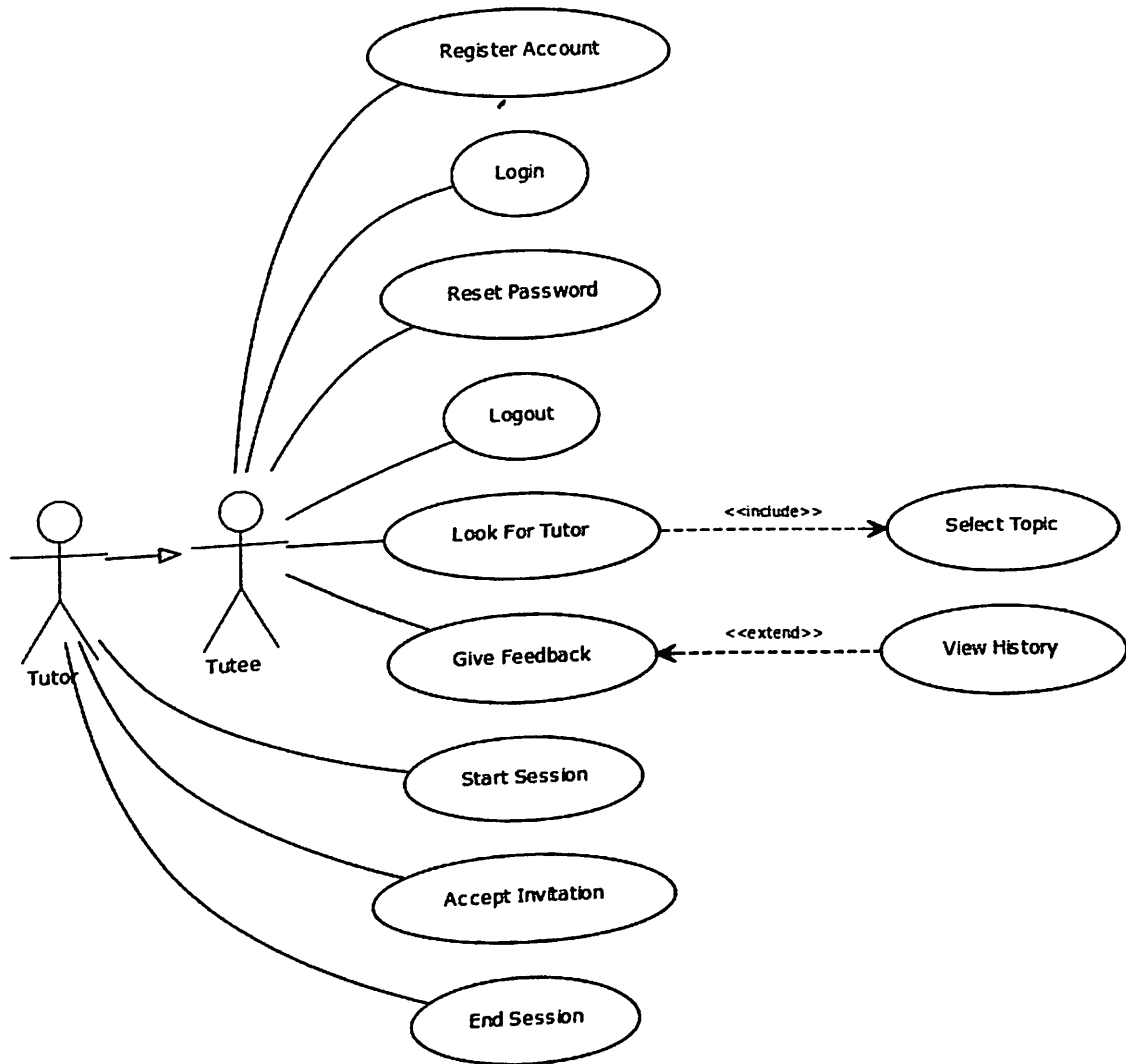


Figure 47. Use case diagram for Go Guro

Use Case Document

Default Aspects of All Use Cases

- Stakeholders:** Project Owners
- Prerequisites:**
- Tutee and Tutor must have mobile data and location services enabled on mobile device.
 - Tutee and Tutor must allow the application to use mobile data and location services on the mobile device when prompted.

UC-00: Register Account

Summary: This use case describes the steps involved in registering for an account.

Priority: Essential

Use Frequency: Always
Direct Actor: Tutee
Prerequisite:

- Tutee must have own, existing email address.

Main Success Scenario:

1. Tutee selects action on main activity.
2. Tutee types in email address.
3. Tutee types in password.
4. Tutee presses "REGISTER" button.
5. System registers Tutee via Firebase.
6. Use case ends successfully.

Alternative Scenario Extension:

- If email address is already registered, use case fails.

UC-01: Login

Summary: This use case describes the steps involved in logging into the application.
Priority: Essential
Use Frequency: Always
Direct Actor: Tutee
Prerequisite:

- Register Account use case must be finished.

Main Success Scenario:

1. Tutee selects action on main activity.
2. Tutee types in email address.
3. Tutee types in password.
4. Tutee presses "LOGIN" button.
5. System authenticates Tutee via Firebase.
6. Use case ends successfully.

Alternative Scenario Extension:

- If password was wrong, use case fails and an error message with "FORGOT PASSWORD?" button are displayed.

UC-02: Reset Password

Summary: This use case describes the steps involved in resetting the Tutee's password.
Priority: Essential
Use Frequency: Sometimes
Direct Actor: Tutee
Prerequisites:

- Register Account use case must be finished.
- Login use case must fail.

Main Success Scenario:

1. Tutee clicks on "FORGOT PASSWORD?" button.
2. System instructs Firebase to send password reset email to registered email address of Tutee.
3. Tutee opens password reset email on an external email client.
4. Tutee clicks link inside email.
5. Tutee types in new password.
6. Tutee pressed "SAVE" button.
7. Firebase changes Tutee's password.

8. Use case ends successfully.

UC-03: Logout

Summary: This use case describes the steps involved in logging out of the application.
Priority: Essential
Use Frequency: Always
Direct Actor: Tutee
Prerequisite:

- Login use case must be finished.

Main Success Scenario:

1. Tutee presses "LOGOUT" button.
2. System kills Tutee's session.

UC-04: Look for Tutor

Summary: This use case describes the steps involved look for a Tutor
Priority: Essential
Use Frequency: Always
Direct Actors: Tutee, Tutor
Prerequisite:

- Select Topic use case must be finished.

Main Success Scenario:

1. System uses Google Maps API and location services to discover nearby Tutors.
2. System plots the location of Tutors onto the map.
3. Tutee presses a button to begin searching for Tutors.
4. System sends invitation to Tutors together with the selected topic.
5. Use case ends successfully.

Alternative Scenario Extensions:

- If one Tutor declines invitation, system looks for another Tutor.
- If no Tutor is found, use case fails and an error message is displayed.
- If Tutee wants to send the Tutor an text message, the Tutee can click the "SEND MESSAGE" button.

UC-05: Start Session

Summary: This use case describes the steps involved in starting a study session.
Priority: Essential
Use Frequency: Always
Direct Actor: Tutor
Prerequisite:

- Accept Invitation use case must be finished.

Main Success Scenario:

1. Tutor presses the "START SESSION" button.
2. System marks the study session as "In

Progress”.

3. Use case ends successfully.

UC-06: End Session

Summary: This use case describes the steps involved in ending a study session.

Priority: Essential

Use Frequency: Always

Direct Actor: Tutee

Prerequisite:

- Start Session use case must be finished.

Main Success Scenario:

1. Tutee presses the “END SESSION” button.
2. System marks the study session as “Finished”.
3. Use case ends successfully.

UC-07: Accept Invitation

Summary: This use case describes the steps involved in accepting the Tutee’s invitation.

Priority: Essential

Use Frequency: Always

Direct Actor: Tutor

Prerequisite:

- Look for Tutor use case must be finished.

Main Success Scenario:

1. System shows Tutee name, profile photo, phone number, and topic on Tutor screen.
2. Tutor presses “ACCEPT” button.
3. Use case ends successfully.

Alternative Scenario Extension:

- If Tutor presses “DECLINE” button, use case fails.

UC-08: Select Topic

Summary: This is an included use case for Look for Tutor use case and describes the steps involved in selecting the topic for the Tutee.

Priority: Expected

Use Frequency: Always

Direct Actor: Tutee

Main Success Scenario:

1. System loads all topics from the database and into autocomplete textbox as suggestions.
2. Tutee selects from the topic suggestions
3. Use case ends successfully.

Alternative Scenario Extension:

- If Tutee did not find the desired topic, the Tutee types in specific topic into a text field.

UC-09: View History

Summary: This use case describes the steps involved in viewing

the study session history.

Priority: Optional

Use Frequency: Sometimes

Direct Actor: Tutee

Prerequisite:

- End Session use case must be finished

Main Success Scenario:

1. Tutee presses "HISTORY" button.
2. System loads all Tutee's study session history into list.
3. Tutee select one item in list to view.
4. System loads information about selected study session.
5. Use case ends successfully.

UC-10: Give Feedback

Summary: This use case is an extension of View History use case and describes the steps involved in giving feedback on study session.

Priority: Optional

Use Frequency: Sometimes

Direct Actor: Tutee

Prerequisite:

- View History use case must be finished.

Main Success Scenario:

1. Tutee selects desired rating level in the 5-star rating bar.
2. Use case ends successfully.

Functional Requirements

- The application must present topics as suggestions in an autocomplete textbox.
- The application must visually show the location of nearby Tutors with a map.
- The application must monitor the study session including its status, duration, and rating.
- The User interface of the application should adapt according to the user's viewing device.
- The application must keep track of the Tutor's rating

Feature Set. Features by Release and Priority

- Release 1.0
 - Essential
 - F-0: Google Maps Integration
 - F-1: Study Session Management
 - Expected
 - F-2: Topic Selection
 - Desired
 - F-3: Attractive User Interface
 - Optional

▪ F-4: Feedback Module

Features

F-0: Google Maps Integration

Priority: Essential
Effort: Days
Risk: Safe
Functional Area(s): Operations
Use Case(s): UC-04
Description: A map is displayed in the application. Tutors are plotted in this map for visual reference of the Tutee.

F-1: Study Session Management

Priority: Essential
Effort: Days
Risk: Safe
Functional Area(s): Administration
Use Case(s): UC-015, UC-06, UC-09
Description: Study session is started and ended in the application by pressing a button. History of study sessions by Tutors and Tutees are also managed.

F-2: Topic Selection

Priority: Expected
Effort: Days
Risk: Safe
Functional Area(s): Administration
Use Case(s): UC-08
Description: Tutees can select topics from autocomplete textbox suggestions, or they can type into the textbox.

F-3: Attractive User interface

Priority: Desired
Effort: Days
Risk: Safe
Functional Area(s): Marketing/Promotion
Description: The application has an attractive User interface that conforms to Google's Material Design language.

F-4: Feedback Module

Priority: Optional
Effort: Days
Risk: Safe
Functional Area(s): Customer Service Support

Use Case(s): UC-10

Description: The application allows the Tutee to rate the Tutor using a 5-point star rating bar.

Non-Functional Requirements

- With a stable client-side Internet connection, the application must respond within five to ten seconds.
- The application must conform to Google's Material Design language.
- Appropriate wordings for notifications and prompt messages must be used.
- Accurate error messages must be displayed.

Environmental Requirements

Hardware Requirements. A mobile device must meet the following requirements:

- Must have screen sizes of at least 426 dp x 320 dp
- Must support any form of 802.11 (for WiFi capability)
- Must support GSM or CDMA (for mobile data capability)
- Must include GPS receiver (for location service capability)
- Must have at least 340MB of memory
- Must have at least 360MB of storage

Software Requirements. The mobile device must have Android 4.0.3 "Ice Cream Sandwich" or higher.

Application Program Interfaces. The following are external, third-party APIs used by the system:

- com.google.firebase:firebase-core:11.8.0
- com.google.firebase:firebase-database:11.8.0
- com.google.firebase:firebase-auth:11.8.0
- com.google.firebase:firebase-storage:11.8.0
- com.firebase:geofire-android:2.3.0
- com.google.android.gms:play-services-maps:11.8.0
- com.google.android.gms:play-services-location:11.8.0
- com.github.bumptech.glide:glide:4.0.0

Data Import and Export Requirements. When a Wi-Fi signal is not available, a subscription to a mobile network that also provides mobile data services is needed.