

## Record of Tasks

Task No	Planned Action	Planned Outcome	Time Estimate (min)	Target Completion Date	Criterion
1	Identify and interview the client	Address the problem that the client is facing.	60	Jun 9	A
2	Write the context of the problem	Establish the problem identified in a clear and concise manner. The problem definition must include who the client is, what the client wants, and indicate a possible solution.	15	Jun 9	A
3	Brainstorm and write a proposed solution for the problem. Rationalize this solution for the client.	Explain in a concise and clear manner the purpose of the project to the client	5	Jun 9	A
4	Write the success criteria of the proposed solution.	A clear set standards to be met by the developer, that suits the client's needs and preferences.	15	Jun 10	A
5	Meet with the client to confirm or revise success criteria.	Confirmed success criteria and ensure it meets the client's standards.	15	Jun 10	A
6	Write a design statement for the proposed solution 'Baybai'	A coherent design statement that outlines the plan for the project.	20	Jun 10	A
7	Draw a system diagram for the proposed app solution.	Have a concrete idea of the software and hardware requirements involved in the development of the application.	10	Jun 10	B
8	Plan & create the wireframe for the	Have a detailed visual representation of the	60	Jun 11	B

	proposed solution.	project which also serves as a guide for the developer during the programming process.			
9	Develop: function `syllabicate`	Syllabicates a Tagalog word based on Tagalog language rules.	80	Jun 31	C
10	Develop: Transcription function	Function that turns a Syllabicated word transcribed to Baybayin script	80	Jul 3	C
11	Develop: HomeScreen GUI (Kivy)	Have a screen that follows the Wireframe Diagram and allows the user to choose between LearnScreen, TranslateScreen, or SavedFlashcardsScreen. Only graphic components, no functionality.	60	Jul 21	C
12	Develop: Translate Screen GUI (Kivy)	Have a screen that follows the Wireframe Diagram and allows the user to input a word to be translated and has a box to contain the transcribed word. Only graphic components, no functionality.	80	Jul 22	C
13	Develop: Intro Screen GUI (Kivy)	Have a screen that follows the Wireframe Diagram and allows the user to see the logo and choose between to register or to log in. Only graphic components, no functionality.	80	Jul 31	C
14	Develop: SignUp Screen GUI (Kivy)	Have a screen that follows the Wireframe Diagram and allows	80	Aug 2	C

		the user to register a new account. Only graphic components, no functionality.			
15	Develop: Login Screen GUI (Kivy)	Have a screen that follows the Wireframe Diagram and allows the user to login to an existing account. Text fields here are for username and password input . Only graphic components, no functionality.	80	Aug 2	C
16	Develop: Learn Screen GUI (Kivy)	Have a screen that allows the user to choose different learning modules organized into levels. Only graphic components, no functionality.	80	Aug 4	C
17	Develop: Flashcards Screen GUI (Kivy)	Have a screen that acts as a template for all the flashcards content. Only graphic components, no functionality.	100	Aug 4	C
18	Develop: Saved Cards Screen GUI (Kivy)	Have a screen that acts as a template for all the saved flashcards content. Only graphic components, no functionality.	20	Aug 5	C
19	Develop: Stats Screen GUI (Kivy)	Have a screen that allows user to see their score. Only graphic components, no functionality.	40	Aug 5	C
20	Develop: Integrate front-end and back-end for IntroScreen, HomeScreen, and	The Kivy and Python aspects of the stated screens are integrated. Example, user is now	20	Aug 5	C

	TranslateScreen.	able to input on the screen, and have the translation returned and reflected on the screen as well.			
21	Develop: Create sqlite database and tables	Create the database tables for users, saved, and userstats.	5	Aug 5	C
22	Develop: database_handler python file	Allows develop to commit and search through a given database.	5	Aug 5	C
23	Develop: Create validation functions	code validation functions that validates email, username, password match and password length that could be used repeatedly in future program development	10	Aug 5	C
24	Develop: Login Screen class in Python	realize user login, validation of username and password match	30	Sep 7	C
25	Develop: Registration Screen class	realize user register, data stored in database	30	Sep 8	C
26	Create learning curriculum based on flashcards using spreadsheet	Using a spreadsheet, create the contents of the flashcards. This csv file has 4 columns: level, index, tagalog, and baybayin.	20 min	Sep 8	B
27	Transfer contents.csv to new sqlite table in baybai.db sqlite database.	Upload csv file to pycharm to automatically transfer from csv to table.	2 min	Sep 10	C
28	Develop Learning class in Python	Methods include going back to home screen and going to Flashcards screen based on the level/module the user wants to learn.	30 min	Sep 12	C
29	Develop: Flashcards class and functions	Accessible from the Learning Screen. On entry, it takes all flashcards from contents table based on the level/module and acts as a template for these contents. Other methods include flipping the card	80 min	Sep 14	C

		(baybayin to tagalog word and vice versa), going to next card, going to previous card, and updating flashcard content.			
30	Develop: Save Card functionality	In Flashcards class, add a method that allows users to save a card and append saved cards to a saved table in the database.	30 min	Sep 16	C
31	Develop: User points system	Have the functionality of saving and retrieving user statistics implemented	45 min	Sep 16	C
32	Develop: Copy translation functionality	Allows the user to click a button and have the transcribed word into the user's clipboard.	10 min	Sep 16	C
33	Develop: Integrate client's chosen font for UI	Integrate Helvetica and Noto Sans Tagalog (Baybayin) using `LabelBase` and method build.	5 min	Sep 16	C
34	Develop: Integrate Learn Screen, Flashcards Screen, SavedCards Screen, and StatsScreen front-end and back-end.	The Kivy and Python aspects of the stated screens are integrated. Example, users are now able to go through the flashcards with the actual contents and expected functionalities.	30 min	Oct 9	C
35	Develop: Check Network connectivity functionality for Network Screen class	Checks if user is connected to network. This acts as a prerequisite for the network function.	20 min	Oct 9	C
36	Develop: Database saved online/in Flask server	Tables in this database are users and posts.	30 min	Oct 9	C
37	Develop: sync current user's logged in credentials to online database	Once user is connected to a network and can connect to the server, the existing login credentials (from offline login) are used to saved user information (name, password, username) to online database.	80 min	Oct 9	C

38	Develop: Posting functionality	Allows user to add post. Post is added to database.	60 min	Oct 9	C
39	Develop: Delete post functionality	Asks user if they're sure to delete post. If yes, post is deleted from database.	20 min	Oct 9	C
40	Develop: Log out	Logs out the user from the application. All userstats are saved.	20 min	Oct 9	C
41	Create test plans	create detailed test plans for the prototype. All types of test including unit test, integrated test, and non-functional test	20 min	Oct 21	B
42	Execute test plans	Execute the test plans that have been done	40 min	Oct 21	B
43	Debugging	Fix issues spotted during testing.	100 min	Oct 22	C
44	Client Evaluation	Evaluated using success criteria and user's preferences. Recommendations and comments are noted.	30 min	Oct 24	E
45	Implement client recommendations	Recommendations given by the client are implemented.	80 min	Nov 2	C
46	Package application and present final output to client	App sent in .apk format.	5 min	Nov 3	D

Table 2. Record of Tasks

## Test Plan

Test No	Test Type	Procedure	Expected Outcome
1	Functional: User Registration (Blank Entry)	1. Run 'baybai.py'. 2. Navigate to Sign-Up Screen. 3. Enter one or more blank entries.	1. Validate appropriate error message(s) for blank field(s). 2. Confirm no data is stored in the database.

Test No	Test Type	Procedure	Expected Outcome
2	Functional: User Registration (Invalid Data [ex. Does not meet password requirements])	1. Run 'baybai.py'. 2. Navigate to Sign-Up Screen. 3. Enter invalid data (e.g., short password).	1. Verify appropriate error message(s) for invalid data. 2. Confirm no data is stored in the database.
3	Functional: User Registration (User Exists)	1. Run 'baybai.py'. 2. Navigate to Sign-Up Screen. 3. Enter a username that already exists.	1. Validate an error message for an existing username. 2. Confirm no duplicate records in the database.
5	Functional: User Registration (Valid Data)	1. Run 'baybai.py'. 2. Navigate to Sign-Up Screen. 3. Enter valid registration data.	1. Check the 'baybai.db' database for a new user record. 2. Ensure user data is correctly stored.
5	Functional: User Registration (Data Security)	1. Run 'baybai.py'. 2. Navigate to Sign-Up Screen. 3. Register a new user. Inspect the 'baybai.db' database.	1. Ensure user data is securely hashed or stored. 2. Confirm sensitive data is not exposed.
6	Functional: User Login (Invalid Credentials)	1. Run 'baybai.py'. 2. Navigate to Login Screen. 3. Enter invalid login credentials.	1. Validate error message(s) for incorrect credentials. 2. Ensure no unauthorized access.
7	Functional: User Login (User Does Not Exist)	1. Run 'baybai.py'. 2. Navigate to Login Screen. 3. Enter a username that doesn't exist.	1. Confirm an error message for a non-existing user. 2. Prevent unauthorized access.
8	Functional: User Login (Valid Credentials)	1. Run 'baybai.py'. 2. Navigate to Login Screen. 3. Enter valid login credentials.	1. Confirm successful access to the app's main features. 2. Verify user actions work as expected.
9	Functional: Flashcards Display	1. Run 'baybai.py'. 2. Access the Learn Screen and select Flashcards. 3. Interact with flashcards.	1. Validate flashcards display correct content and can be flipped. 2. Verify starring/unstarring cards updates saved state.

Test No	Test Type	Procedure	Expected Outcome
10	Functional: Syllabication Function	1. Run 'baybai.py'. 2. Test the 'syllabicate' function using a dataset of Tagalog words.	1. Ensure correct syllabication of Tagalog words based on the dataset.
11	Functional: Tagalog to Baybayin Transcription	1. Run 'baybai.py'. 2. Navigate to Translate Screen. 3. Enter Tagalog text for translation.	1. Confirm accurate Tagalog-to-Baybayin translation. 2. Verify translated output is displayed correctly.
12	Functional: Wrong input in Tagalog to Baybayin Transcription	On Translate Screen, input an English word, any word not in Tagalog, or a word with a letter not in the Tagalog alphabet (ex. 'Z')	A popup shows up telling the user to try again and make sure to input only a Tagalog word.
14	Functional: Copy Translation	1. Run 'baybai.py'. 2. Navigate to Translate Screen. 3. Request a translation and use "Copy Translation" option.	1. Confirm "Copy Translation" copies Baybayin text to clipboard. 2. Verify copied text is correct and complete.
13	Functional: Save a Card	1. Run 'baybai.py'. 2. Access the Flashcards Screen. 3. Star a flashcard to save it. 4. Navigate to Saved Cards Screen.	1. Ensure starring a flashcard saves it successfully. 2. Verify the saved card appears in Saved Cards.
14	Functional: Remove Saved Card	On the Saved Cards Screen, unsave or untoggle an already saved card.	The card will be removed.
15	Functional: Empty saved cards	Register a new account. This makes sure that account has no saved cards.	A popup shows up saying there are no saved cards as of the moment.



Test No	Test Type	Procedure	Expected Outcome
15	Functional: Network Connectivity	1. Run 'baybai.py'. 2. Use the app under various network conditions (online/offline). 3. Observe app's response to network changes.	1. Validate app accurately detects and reports online/offline status. 2. Confirm relevant user prompts and interactions.
16	Functional: Add Post Functionality	1. Run 'server.py'. 2. Run 'baybai.py'. 3. Access the Network Screen. 4. Create and submit a post.	1. Ensure successful posting of user-generated content. 2. Confirm posts appear on the Network Screen.
17	Non-Functional: Code Clarity	1. Review the 'baybai.py', 'baybai.kv', and 'server.py' codebase. 2. Assess code clarity, organization, and comments.	1. Evaluate codebase for readability and maintainability. 2. Identify and address areas for improvement.
18	Client Testing	1. Prepare test devices (emulators, physical devices). 2. Install 'baybai' app package on test devices. 3. Perform functional and usability tests.	1. Verify app functions as expected on different devices. 2. Confirm usability on various screen sizes and resolutions.
19	Client Testing After Updates	1. Release app updates. 2. Inform test users to update 'baybai'. 3. Re-run functional and usability tests.	1. Ensure updated app versions perform as expected. 2. Verify compatibility with previous data and settings.
20	Integration: Application Package Testing	1. Package the 'baybai' app for delivery to the client. 2. Install and test the packaged app on client devices.	1. Confirm successful installation and functionality on client devices. 2. Ensure the app operates seamlessly in the client's environment.

Table 3. Test Plan