MOBDEVE Machine Project

AY2021 T3

Students are assigned a term long project where they are to design, propose, and develop a mobile application that can run on an Android device. The following specifications detail what is expected of the students.

General Notes

- The Android application will be built using Java.
- The machine project is meant to be done in a group.
  - Typically, a group may consist of 2 to 3 members; however, your instructor reserves the right to modify the group size according to their own prerogative.
  - No consideration will be given to those who would want to form groups bigger than the max size agreed upon.
- If there are any questions or need for clarification of certain points, approach your teacher immediately.

Application Requirements

The application must contain at least two (2) of the following services*:

- SMS/Telephony
- Local database (SQLite)
- Geolocation/Maps
- Touch gestures (non-native to the Views used)
- Camera Usage
- Canvas (for games) or Graphics-Related Applications
- Motion sensors (accelerometer, gyroscope, etc.)
- Service (background running) with or without Broadcast receiver
- Usage of 3rd-party APIs**

* These services/APIs must have a clear purpose in the app, and not just added for the sake of fulfilling the Requirements.
** Not all APIs will count. Please make sure to consult with your instructor.

If the app doesn't have two of any of those listed above, they need to at least be connected to a remote server (e.g Firebase) for the developed app. The group is expected to develop the back-end system in the remote server/Firebase.

Project Proposal (due: Aug 6, 2021)

The proposal must contain an overview of the application, a list of services / APIs planned to be used, and a list of functionalities. While the requirements here are rather straightforward, please note the following:

- The description must properly capture the idea of the application.
- Each service / API must be accompanied by a short statement describing why the service / API was chosen for the app. Inclusion of a service / API must be aligned with the overall theme / goals of the application.
- Describing the functionalities can be done in many ways. One way would be to describe what a user can do in each screen / Activity. Another way might be to describe all individual functions and state which screen they're projected to be associated with. The description doesn't need to be too specific in terms of how functions would be implemented, but it is encouraged to be as specific as possible. All functionalities should support the overall description of the application.

While use of services / APIs is a quantified requirement, stating the functionalities will scope the project and will also be evaluated in terms of whether to accept or modify the proposal.

The project proposal is graded and is worth 5% of the total grade for the course. For the template and manner of grading of the proposal, please refer to the succeeding pages.

Beta Demo (due: Aug 27, 2021)

By the end of the tenth week, the group is expected to have their app's activity layouts completed and have either (a) the database designed correctly with all the database helpers implemented or (b) at least one app feature (API) properly implemented. If the group aims to implement a database, the beta demo will not require connection of the database to the user interface. The requirements stated are defined loosely to achieve generalization across machine projects; however, your instructor reserves the right to require more or less of a requirement depending on the difficulty / complexity of the database design or feature / API implementation. It would be good to consult with your instructor regarding expectations for the beta demo.

As proof of the app's progress, a demonstration is expected to be conducted. First, the demonstration should clearly show the different layouts of the application. Logic within each activity does not need to be implemented, but the app should be able to move from activity to activity. If a database is to be presented, the demonstration should walk through the structure of the DB and quickly show how it was implemented in code. If a feature / API is to be implemented, the demonstration should show the feature working within app and quickly show how it was implemented in code. Discussion of the code doesn't need to be in detail, but enough to demonstrate your understanding of the app. Any deficiencies / shortcomings (e.g. bug of a feature, incomplete layout) should also be highlighted in the demonstration.

The demonstration may be conducted synchronously (i.e. live through a conferencing tool) or asynchronously (i.e. pre-recorded video) depending on your instructor's prerogative.

In addition to the demonstration, the group is expected to submit a link to their respective version control repositories. If the repository is hosted privately, kindly ensure that your instructor has access to it. The link submitted will be used to monitor the progress off your application. Usage of version control is not part of the rubric for the beta demo; however, it is still a requirement. More details of this are found below.

The beta demo is worth 10% of the total grade. For more information on how the demo will be graded, please see the rubric below.

Submission of Final Requirements (due: Sep 14, 2021)

The group must submit the following via Canvas:

- Zipped file of the Android project
  - While your instructor will have access to the repository, this requirement is so that there is a copy of the submission that will persist if ever anything happens to the repository. Please make sure that the copy submitted is the latest version aligned with the latest commit prior to the deadline.
- Final Demo
  - The group is asked to conduct a demonstration of their application highlighting the application fulfilling the project proposal. Please refer to the rubric below for what will be expected of your application. The demo may either be conducted live or asynchronous (through a video). Discretion is left to the instructor.

Please make sure that your services / API can run on machines / phones outside of your test environment. If you're using a remote server, please make sure to allow access to appropriate requests until the release of grades so your instructors can properly evaluate your application.

For the manner of grading of the final submission, please refer to the succeeding pages.

Project Management

As previously mentioned, the Android project must be maintained through a version control repository. Each repository must show continuous progress and not a single push at the end of the term. The project itself must also adhere to proper coding standards. Project management is part of the rubric for the final requirements submission.

Bonus Points for Kotlin Implementation

10 points will be awarded to the final requirements score if the application was built completely with Kotlin instead of Java.

APPENDIX A – Template for Project Proposal

	MOBDEVE Machine Project
	BSCS-ST[1], BSIT[2], BS<specialization><superscript>
	iVolunteer Mobile App
	Rizal, Jose[1] | Bonifacio, Andres[2] | Aguinaldo, Emilio[1]

Description

The application is a volunteer registration app for iVolunteer. iVolunteer is in partnership with up to 50 NGOs and allows its users to register for a volunteer program on their app. The user is allowed to log-in, see available volunteer programs, and filter through the volunteer programs by time, venue, and advocacy. When a user joins a program, an event will be added to the user's default calendar app and will notify the user 2-3 days before the event. On the volunteer site, the user has the option to "check-in" for the program. A feedback will be asked from the user 2-3 days after the event, asking the user to share his photos and experience on Facebook.

Services / APIs

Web Server (Firebase)

- To store / verify user credentials and available programs
- To handle scheduling of event notification

Facebook API

- Write access to a user's FB account to share experiences

Geolocation

- Used to verify if the current location matches the location of the program

Functions

Function	Description
Register	The user must first register an account before accessing the other features of the app. Users are required to give their full name, birthday, email, and password.
Log-in	The user must log-in before joining/viewing a program. This requires the user to enter his/her registered email and a password.
View Program List	The user can view the list of available of volunteer programs, each program containing basic details of the event. The list will be provided via a remote web server.
View Program Details	Once a user selects a program, the app will provide the complete details of the event.
Filter Programs	The user can filter through the programs based on date/time, advocacy (e.g. education, housing), and location.
Register/Join Program	The user can join a program (log-in is required if the user hasn't log-in yet). This will update the remote database.
Program Notification	The app will notify the user three days, and a day before a registered event.
Check-in	The user can "check-in" when he is on-location. The app will verify the user's participation by checking if the location and time matches.
Feedback Notification	The app will notify the user to share his experience days after a program.

Feedback	The app will allow the user to share his photos and experience on Facebook.

APPENDIX B – Project Proposal Rubric

Total of [100 pts]

Project Format (20 points)

Was the template properly followed? Were all parts filled out?

Proposal Overview (20 points)

Is the proposed project sound? Is it feasible and realistically do-able given the time frame?

Proposal Technicalities (30 points)

Are the software requirements fulfilled? Are the required libraries and technologies adequate for the project proposed?

Proposal Features (30 points)

What are the features of the application? Is it appropriate for the core of the application? Is it usable and practical for the purposes of the app?

APPENDIX C – Beta Demo Rubric [100 pts]

Expected Output

- Complete activities to be used in the final app, with all the views included
- Database has already been designed (in-app, or server side). Tables and basic helper methods (CRUD) are created. If no database is needed for the app, at least one app feature has been explored (gestures, camera, etc.)
- UI need not be connected to the database yet

Project Goals (50 points)

Can I use the app for purpose X? Are the tasks that I'll perform in working condition? Are the tools I need to achieve the app goal complete?

Aspect	50 - 40 points	39 – 16 points	15 – 0 points
Implementation	The database has been designed correctly with all the database helpers implemented. **OR** At least one app feature (API) has been developed	Around 80%-25% of the database and helper methods exist **OR** One app feature (API) has been implemented partially (roughly 80%-25%)	The database design isn't appropriate **OR** No features were implemented **OR** Less than 25% of the database/feature is present

User Experience (25 points)

Can I easily use the app without any aid? Is the design meant to be friendly to the audience it's intended for? Can I get things done in a few types and clicks? Is navigation easy to follow?

Aspect	25 – 20 points	17 – 12 points	10 – 0 points
Content Organization	Presentation of info (access and manipulation) is clear, and is appropriate for the goal/theme of the app.	Content (access and manipulation) could sometimes presented in a better way that matches the project goals/theme	Content could not be accessed or manipulated, or was better conveyed without the app
Navigation	App is easy to navigate, it is clear to the user where they are, and what steps to take to achieve a task	User is sometimes confused how a task can be achieved	User cannot navigate without the help of the developer

View Titles	Users are easily guided by the text provided.	User can misconstrue text provided; App has spelling or grammar mistakes	User cannot understand text provided; Spelling and grammar mistakes abound

Visual Design (25 points)

Is the app easy on the eyes? Does it match the aesthetic of its content? Are the icons clear and appropriate? Are the words used appropriate for its content?

Aspect	**25 – 20 points**	**17 – 12 points**	**10 – 0 points**
Message (Strength)	App UI design is clear and concise	App design is present, but not always clear	App design is not present
Message Matches Project Brief	The app design matches the project's goals and theme	App design is sometimes not aligned to the project	App design is not appropriate for the project
Graphics	Appropriate graphics were used when necessary, icons are easy to understand; If difficult to understand, icon is accompanied with supporting text	Graphics are related to the theme, but are of not good quality	Graphics are not used, or only made the user distracted

APPENDIX D –Final Demo Rubric

Total of [100 pts]

Project Goals (50 points)

Can I use the app for purpose X? Are the tasks that I'll perform in working condition? Are the tools I need to achieve the app goal complete?

Aspect	45 – 50 points	44 – 40 points	35 – 25 points	23 – 13 points	12 – 0 points
Message Matches Project Contract	The group developed all the features and fulfilled objectives outlined in the project brief. The group also added extra features to support the objectives.	The group developed all the features and fulfilled all objectives outlined in the project brief.	The group developed most of the features and fulfilled most objectives outlined in the project brief.	The group developed the core features and only fulfilled a few of the objectives outlines in the project brief.	The group did not develop the features or fulfilled any objective in the project brief.
Content Organization	The group organized the information according to the project objectives and exceeded in its effort to display content.	The group organized the information according to the project objectives.	The group made a minimum effort to organize the information according to the project objectives.	The group displayed the information, but organization is lacking.	The group did not display any information at all.
Fulfills Project Goal	The group completely fulfilled the project goal.	The group mostly fulfilled the project goal. (>75%)	The group somewhat fulfilled the project goal. (>50%)	The group only fulfilled a part of the project goal. (>25%)	The group did not try to complete the goal at all.

User Experience (20 points)

Can I easily use the app without any aid? Is the design meant to be friendly to the audience it's intended for? Can I get things done in a few types and clicks? Is navigation easy to follow?

Aspect	20 – 17 points	16 – 13 points	12 – 7 points	8 – 5 points	4 – 0 points
Content Organization	The group made an exceptional consideration and execution to organize the content of the app for the people whom it is intended.	The group organized the content of the app for the people whom it is intended.	The group put information in the app but did not organize it for the target audience.	The group only provided the minimum amount of information and organization for the target audience.	The group did not organize the information for the target audience.
Navigation	A user easily navigates without help.	All navigation items are clearly labeled and consistently placed allowing	Many navigation items are labeled, allowing the reader to	Some navigation items are labeled, allowing the reader to move	Navigation items do

		the user to easily move from a view to related views and take the reader where s/he expects to go. A user does not get confused.	move from a view to related views. A user rarely gets confused.	from a view to related views. A user sometimes gets confused.	not clearly describe where the reader will end up. A user typically feels confused.
View Titles	All views have clear titles.	Many views have clear titles. (>75%)	Some views have clear titles. (>50%)	Few views have clear titles (>25%)	View titles are missing.

Visual Design (20 points)

Is the app easy on the eyes? Does it match the aesthetic of its content? Are the icons clear and appropriate? Are the words used appropriate for its content?

Aspect	**20 – 17 points**	**16 – 13 points**	**12 – 7 points**	**8 – 5 points**	**4 – 0 points**
Message Strength	Intended message is clear and concise		Intended message is present but not clear.	Intended message is obscure.	Intended message is missing.
Message matches project brief	Intended message fulfills all goals and objectives as outlined in project contract.	Intended message fulfills most goals and objectives as outlined in project contract.	Intended message fulfills some goals and objectives as outlined in project contract.	Intended message fulfills few goals and objectives as outlined in project contract.	Intended message is unrelated to goals and objectives as outlined in project contract.
Graphics	Graphics are related to the theme/purpose of the app, are of excellent quality and enhance reader interest or understanding.	Graphics are related to the theme/purpose of the site, are of good quality and enhance reader interest or understanding.	Graphics are related to the theme/purpose of the site and are of good quality.	Graphics are related to the theme/purpose of the site but are of poor quality.	Graphics seem randomly chosen, are of low quality, or distract the reader

Project Management (10 points)

Can I easily use the app without any aid? Is the design meant to be friendly to the audience it's intended for? Can I get things done in a few types and clicks? Is navigation easy to follow?

Aspect	**10 – 9 points**	**8 – 7 points**	**6 – 4 points**	**3 – 1 points**	**0 points**
Proper Usage of Version Control	The group consistently used version control to track progress and collaboration is evident.	The group used version control, but it wasn't used consistently.	The group used version control, but it was used in bursts or progress to the project was mainly led by only one member of the group.	The group had minimal use of version control, or its use was mostly just prior to submission. It also does not properly show the contributions of each member.	No version control was used by the group or no evidence was provided.
Coding Standards	The project adheres to coding standards. The code is well organized and very easy to follow. Comments are present and aid to understanding. Naming convention of files, methods, and variables is consistent and well thought through.	There may be some minor failures to adhere to standards. Indentation or naming convention may be inconsistent. Naming convention of variables, methods, or files may not immediately give insight into the purpose of the entity.	There are some moderate violations of the standards. The code may be difficult to read in places, or some of the comments may be unclear. However, the code is still generally readable and there is enough commenting, good enough variable names, etc. that someone could still figure out what's going on without too much difficulty. Functions and the basic algorithmic structure are still generally appropriate.	There are major problems with the program's design or coding style that would interfere with its comprehension, reuse, or maintenance. A good number of comments are unclear, variable descriptions or names do not reveal the purpose of use. Code may be poorly formatted. However, there is still some value apparent in the code, perhaps in its overall decomposition into different methods or algorithmic blocks.	Comments are generally missing. The code is difficult to read and it's design is difficult to follow. Methods may include mixtures of different functionalities that don't belong together. However, there is at least one redeeming quality that could allow someone familiar with what it is supposed to do to understand some aspect of it.