

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaclST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

1 Overview

In this project you will learn how to develop a non-trivial mobile application for the Android operating system. This application will explore several key aspects of mobile development, including location awareness, judicious use of limited resources, and social behavior. Towards this end, you will be developing a simple but powerful pharmacy management app: PharmaclST.

This app gives mobile support to pharmacies and their users. PharmaclST should allow users to find pharmacies and medicines in their surroundings, view information about pharmacies and their available stocks, search for medicines at those pharmacies and reserve medicines for future purchase.

The project will be built as the sum of a set of mandatory and additional components and features. The idea is that everyone implements the mandatory features (worth up to 75% of the grade), and then selects a combination of additional features that add up to complete the grade at 100%. Grades from the optional features are allowed to accumulate beyond 25% (and compensate for any limitations in the mandatory part of the submission), though the final grade is capped at 100%.

2 Mandatory Features

The mandatory features include the core library management functionalities. Though the exact UI design is up to you, the app should support the following features:

- Allow users to create a username in the application/system.
- Show a map with pharmacy locations:
 - The map can be dragged, address searched, or centered on current location;
 - Favorite pharmacies have a different marker;
 - Tapping a marker goes to a pharmacy information panel;
- There should be an option to add a new pharmacy with:
 - Name;
 - Pick location on map, use address, or current location;
 - Take picture;
- Find medicines (including at least a sub-string search), providing the closest pharmacy with the searched medicine;
- Pharmacy Information Panel:
 - Show name, location on map, and picture. Button to navigate there;
 - List available medicines;
 - Button to add medicine stock (scan barcode) or create medicine if code unknown:
 - * Name;
 - * Box photo (from camera or file);

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaIcST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

- * Quantity;
- * Purpose/Preferred Use;
 - Button to purchase/reduce stock(scan barcode);
 - Button to add/remove pharmacy from favorites;
 - Tapping medicines opens medicine information panel;
- Medicine Information Panel:
 - Show name and picture;
 - Button to get notification when available in favorite pharmacy;
 - List pharmacies where available, sorted by distance;

2.1 Back-end

PharmaIcST supports a number of features that build on explicit data sharing and crowd-sourcing among multiple devices. To enable such functionality you will need some sort of back-end service that holds and processes shared data (e.g. pharmacy information) and that each device communicates with to synchronize its state.

How you implement this back-end service is not the focus of this class but is nonetheless necessary for the project to work. Feel free to implement your own server as you see fit (e.g. using Java RMI, gRPC, or a RESTful service), or otherwise use an existing server software or database. The server can be kept simple for debug and development purposes and can hold data in memory alone (no need for persistence). In any case, be prepared to justify your choices.

2.2 Resource Frugality

When using real-time data in mobile applications, developers often have to contend with a trade-off between timeliness and efficiency. It's nice if the user gets notified the moment a pharmacy's status changes, but constantly polling the server can drain the battery and use up the user's metered data. In PharmaIcST, pharmacy and medicine information should sync across devices in a timely manner while also using the network efficiently. When the user is actively viewing a pharmacy or medicine, ensure that any new content shows up quickly. If the user disengages from the application, use more efficient messaging to save network resources, even if at the expense of increased latency.

It's also important to avoid using resources unnecessarily. Avoid wasting resources by only downloading data related to UI elements as they become visible to the user. For example, when a user searches for medicines using a search filter, search results can be downloaded only as scrolling requires. Furthermore, search filters should be applied on the server side to reduce data transmission.

Particularly large content can be further optimized to avoid costly metered data. Photos, e.g. medicine boxes, can represent a hefty data cost so, to optimize network usage,

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaclST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

show a placeholder for them when the user is on a metered connection, retrieving the image only when the user taps it. If, on the other hand, the user is on WiFi, automatically retrieve photos when visible.

2.3 Context Awareness and Privacy

The PharmaclST application should be aware of its location and should automatically open free pharmacy information panels when within range of the pharmacy (e.g. 100m).

2.4 Caching

Often users have only spotty data-connections with metered data. As such, communication between the PharmaclST application and its back-end server should be optimized to use the network judiciously and to compensate for short term outages. On the one hand you should avoid downloading the same content multiple times when it could reasonably be avoided, on the other you want to minimize disruption during a momentary outage.

To address this challenge, use a cache to store content as you retrieve it from the server. With this in place, repeated downloads of images are minimized and any content recently viewed will be available offline if needed.

Further optimize your cache through careful pre-loading when the user connects to WiFi. As WiFi data is virtually free, use the opportunity to load the most recent content (the content immediately visible when opening the pharmacy information panel without scrolling). This way, later when the user no longer has WiFi, they can still browse a large set of nearby pharmacies with minimal data usage.

3 Additional Components

In this section we list a series of features that can be combined to add value to PharmaclST. Each feature lists the grading percentage it is worth, reflecting the relative difficulty in implementing it.

The project core (described above) is worth 75%, which leaves at least 25% for these additional components. Select a combination of these features that adds up to or exceeds that threshold.

Many of these features can be naturally integrated with each other and gain from being implemented in such a manner. Implementing your selection of additional features in such a cohesive manner is encouraged.

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaIST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

3.1 Securing Communication [5%]

Sending data across a network in the clear poses a significant risk to users. A malicious party can eavesdrop the data being transmitted and use it to infer all sorts of private information, including e.g. a user's location. Unprotected transmissions can also be modified in transit, feeding users fake data that can influence their decisions (e.g. putting words in people's mouths) or even put them in danger (give them directions to the middle of a construction site).

To avoid such scenarios, upgrade the PharmaIST infrastructure to secure all communications between the mobile application and the back-end server to use SSL. Also, do be careful with how you manage your certificates to be sure the client only communicates with your authorized PharmaIST server, preventing an untrusted party from conducting a man-in-the-middle attack.

3.2 Meta Moderation [10%]

Another way to attack the integrity of PharmaIST data is for the attacker to simulate one or more valid users and to then simply feed bad data into the system. There is no easy way to completely solve this problem but a meta-moderation system raises the bar.

Create a mechanism whereby users can flag suspicious pharmacies. As users flag entities, keep track of how many times each entities has been flagged by distinct users. When a pharmacy is flagged, immediately hide it from the flagger's view, providing immediate feedback that their concern has been registered. As more users flag that pharmacy, hide it from everyone once a given threshold has been surpassed.

Also keep track of how many pharmacies have been blocked in this manner for each user. If a particularly obnoxious user has had enough of their pharmacies flagged, suspend her/him from the system.

3.3 User Ratings [10%]

A good way to guide readers to good pharmacies is to allow for user-submitted pharmacy ratings. Allow users to rate pharmacies with 1-5 stars. If a user tries to submit a new rating, it should overwrite the previous one. If ratings are supported, the pharmacy information panel should include a histogram of all ratings.

3.4 User Accounts [10%]

The project core does not include a login/logout procedure so users can be tracked via simple GUIDs. This simplifies the user experience, as users need not setup an account before using PharmaIST but it also makes it more difficult to keep multiple devices consistent.

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaclST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

Allow users to create accounts so that they can login from multiple devices and seamlessly keep their list of pharmacies in sync. Include Login/Logout buttons in the context menu. Multiple devices logged in for the same user should automatically sync the same pharmacies, providing a seamless cross-device experience. User accounts should be optional, so users can start using the app without an explicit account and later upgrade to using an account if they want.

3.5 Social Sharing To Other Apps [5%]

Another important aspect in mobile development is social sharing. It's often convenient to share information with friends in context and directly from within an application, without needing to open a separate communication app to do so.

Add the option to share content from PharmaclST (e.g. good pharmacies, available medicines) with other apps, e.g. common social media (e.g. Facebook, Twitter) and communication applications (e-mail).

3.6 Localization [5%]

Users are diverse and multi-lingual and PharmaclST should reflect that. Localization – also called L10n – is a collection of tools and techniques that allow different users that speak different languages to use the application and share data without barriers.

Translate all static strings presented to the user into at least two languages (e.g. English and your own native language) and store these strings in such a way that adding new translations is easy and does not require refactoring the application. A database of static strings works well for menus and buttons, but cannot handle user submitted content. Use the Google Translate API, or equivalent, to translate user submitted text messages to their chosen language. When showing text like this, clearly indicate that the new text is a translation and add the option to show/hide the original.

3.7 UI Adaptability: Rotation [5%]

Users don't always use their phones with the same orientation and can rotate to view content either vertically or horizontally, as they see fit. Allow PharmaclST to adapt to these circumstances and make sure all screens show correctly given the current orientation. Make sure the user can experience the application equally well with the phone oriented vertically or horizontally.

3.8 UI Adaptability: Light/Dark Theme [5%]

Though many applications default to a light theme with dark text on a light background, many users prefer a dark theme which is less harsh on the eyes, especially at night, and

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmaclST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

mitigates burn-in on OLED screens. The Android operating system has native support for dark themes and lets the user pick between a light and dark mode within the phone settings. Applications can then adapt accordingly, using an appropriate theme to ensure a consistent look-and-feel with the rest of the phone UI.

Implement both a light and a dark theme in PharmaclST. Make sure the user can experience the application equally well in either mode.

3.9 Recommendations [10%]

Finding relevant pharmacy products that match users' needs is often a challenge but users typically share common needs which can be crowd-sourced and used to suggest new medicines that might interest them. The family of algorithms that power these suggestions is known as *recommender systems* and they power all kinds of recommendations, from e-commerce ("people who bought X also bought Y."), to dating apps, and much more.

Build a simple recommender system for medicines and add a new dialog to PharmaclST that suggests new medicines/products to the user that may interest them. This isn't an ML class so we can use a simple crowd-sourcing algorithm on the back-end: for every pair of medicines/products, track how many users have bought both of them. Based on medicines bought by the user, recommend others. Sort the medicines by this metric and present to the user the top few (enough to fit the screen), which are most likely to be the ones they will be interested in.

4 Alternative Projects

Groups have the option to develop a different project if they so wish and with faculty approval. Consider this option if you already had a project in mind, whether for some collaboration with industry, to explore an idea for a start-up, or as a passion project. We might need to make adjustments to manage complexity, given the time-frame and effort allocated to the course. There are also some features that we consider essential to mobile development and we may need to add or adjust functionality in your idea for the sake of the class project. You can also propose additional features to this project, if you so wish. Start a discussion with the faculty and seek approval as early as possible if you wish to try this option.

5 Grading Process and Milestones

Projects will be evaluated on a variety of dimensions. The most important factors are: the degree to which the specification is implemented, the technical quality of algorithms and

Mobile and Ubiquitous Computing — 2023/24		Assignment:	Project
Mobile Application Development for Android		Issued:	2024-04-15
PharmacIST		Checkpoint:	2023-05-10
Authors:	Prof. João C. Garcia	Due:	2023-05-31
		Version:	1

protocols, and resource efficiency decisions. We will also assess the responsiveness and intuitiveness of the application interface. This is not a course in graphic design so, beyond basic utility and effectiveness, GUI aesthetics will not be graded. Do not invest too much time in creating pretty icons or other superficial assets and focus on making information accessible to the user. Consider using Unicode characters in place of hand drawn icons, when appropriate.

There are two important project milestones:

May 10th: Project Checkpoint The project checkpoint is optional and can only improve your grade. If the checkpoint grade is higher than the final submission grade, it will contribute 20% towards the final project grade. If it is lower, only the final submission/demo/discussion grade will be considered. : Students should submit their current prototype so that they can receive feedback on the architecture and status of the prototype and their checkpoint grade. For the checkpoint submission, groups should have the mandatory functionality ready, as well as a plan for the set of additional features you intend to implement. Consider including an Activity Wireframe and a prepared list of questions to focus the feedback.

May 31st: Project Submission Students should submit a fully functional prototype and a final report. All source code and the report must be submitted through the course Fénix website. A template of the report will be published on the website. The report must describe what was and was not implemented, and is limited to 5 pages (not including the cover). The cover must indicate the group number, and the name and student ID number for each of the group's elements.

6 Collaboration, Fraud, and Group Grading

Student groups are allowed and encouraged to discuss their project's technical solutions without showing, sharing, or copying code with other groups or any other person, whether attending the course or otherwise. Fraud detection tools will be used to detect code similarities. Instances of fraud will disqualify the groups involved and will be reported to the relevant authorities. Furthermore, within each group all students are expected to have a working knowledge of the entire project's code.