Week 6 Blog

Sree Ram Boyapati	a1775690@student.adelaide.edu.au
Course	Comp Sci 7092 - Mobile and Wireless Systems
Group	CovidGuard-F
Supervisor	Zach Wang

What have I done this week?

I have figured out a few key parts for client-server communication. Async Task was being deprecated in Android API 30 and Google suggested that we use *java.util.concurrent* package for asynchronous requests.

Some of the challenges I faced -

1. Few requests were taking a lot of time and were a huge penalty hit for signing up the user.
2. Who owns the execution? Should the activity provide the asynchronous executor and manage the connection pool and let the service handle the abstractions and raise an IOException? This is important for fine-grained parallelism.
3. How to pass context to the app executors after implementation asynchronously? Presently, We are making a blocking operation while waiting for the response from the slow server. *ListenableFuture* interface from Google Guava library can solve the problem, however, it is quite huge and may increase the APK size which is already 14 MB. **We need to use ProGuard optimization for the final APK bundle.**

Outcomes:

1. Successful Registration of Token in the backend. PR: https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/pull/23
2. SQLite Interfaces for persisting data of RPI and TEK in the SQL database. PR: https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/pull/24
3. A global pool of threads for handling network and diskIO operations. https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/blob/master/client/CovidGuard/app/src/main/java/com/project/covidguard/AppExecutors.java

Issues Identified

1. https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/issues/26 - To reduce the blocking time of the registration request.
2. https://github.cs.adelaide.edu.au/2020-Mobile-and-Wireless-Systems/CovidGuard-F/issues/25 - To encrypt data at rest.

Token Registration Flow

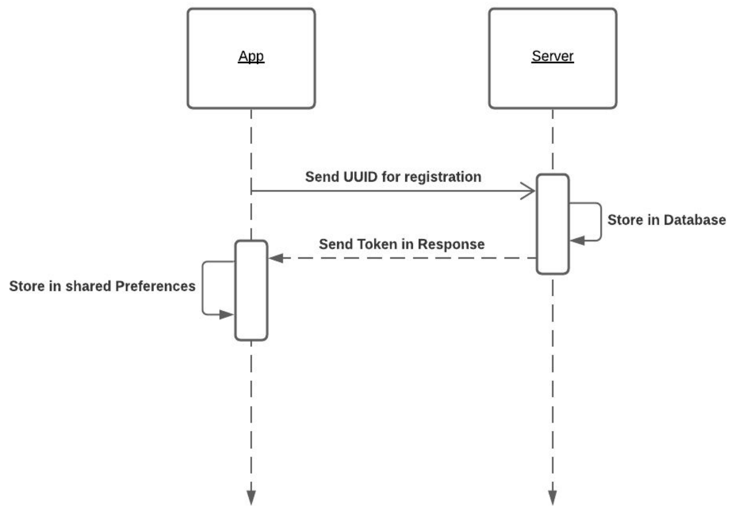
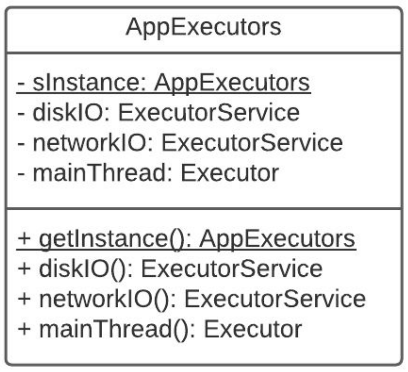


Figure 1: Registration Overview


AppExecutors
- sInstance: AppExecutors
- diskIO: ExecutorService
- networkIO: ExecutorService
- mainThread: Executor
+ getInstance(): AppExecutors
+ diskIO(): ExecutorService
+ networkIO(): ExecutorService
+ mainThread(): Executor

1. App sends an asynchronous request to */register_uuid* with the UUID.
2. The server registers the UUID and generates a token for authentication purposes and stores it in database and sends the token to the user.
3. The user stores the token in shared preferences.

To speed up the app response times for the first request - I have added warmup handler[1] at the server to warmup database connections after the instance is initialized.

To make asynchronous requests:

1. I defined a global application class called app executors that encapsulated executor services for diskIO, networkIO and MainThreadHandler (UIThread) in android and initialised them during application launch.

[1] "Configuring Warmup Requests to Improve Performance." 14 Jul. 2020, https://cloud.google.com/appengine/docs/standard/java/configuring-warmup-requests. Accessed 6 Sep. 2020.

App Storage Layer

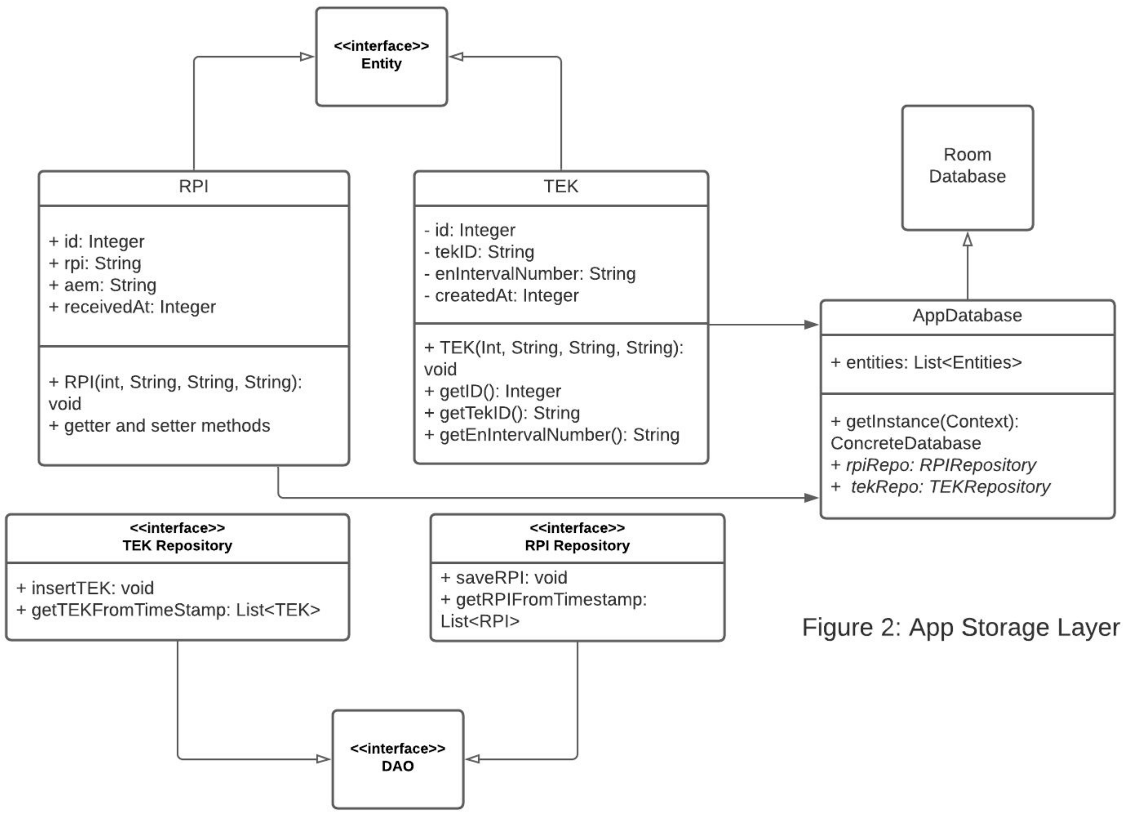


Figure 2: App Storage Layer

1. Entities represent tables in the SQLite database
2. Repositories are data access objects (DAO) to the tables and have definitions of queries that will be run.
3. App Database is a singleton that is initialised and extends AbstractDatabase which is part of Room[2] Library.

[2] "Room Persistence Library | Android Developers." https://developer.android.com/topic/libraries/architecture/room. Accessed 6 Sep. 2020.

Paper Review #1

Vukolic, M.. “On the Interoperability of Decentralized Exposure Notification Systems.” ArXiv abs/2006.13087 (2020): n. pag.

Why have I selected this paper?

GAEN API is loosely based on DP3T[3] protocol and there are many decentralized exposure notification systems. In their analysis of swissCovid app[4] , Prof.Vaudenay underscored the control of google and apple in exposure notification systems. He criticised the fact that GAEN code is not opensource by design and in fact, I could only find the snapshot of code[5] which is part of google play services on Github. There has been no independent audit of the code.

GAEN platform advises governments to adopt to one single system for all their citizens and GAEN API is only open to authorized public authorities. By its restrictive policy, GAEN is taking away the choice of exposure notification platform to use from its users. However, there are many decentralized applications today. After COVID-19, contact tracing may have multiple use-cases and interoperability with other devices like wearables can give users a choice on what applications they can run.

I have studied this paper to make our app and server compatible with other decentralized backends. The paper additionally explores cross-country infections and manage exchange of information maintaining user privacy.

[3] "Decentralized Privacy-Preserving Proximity Tracing - Wikipedia." https://en.wikipedia.org/wiki/Decentralized_Privacy-Preserving_Proximity_Tracing. Accessed 6 Sep. 2020.
[4] "[PDF] Analysis of SwissCovid | Semantic Scholar." https://www.semanticscholar.org/paper/Analysis-of-SwissCovid-Vaudenay-Vuagnoux/ca590815353c6b3469e2071ceda092900a127c79. Accessed 6 Sep. 2020.
[5] "google/exposure-notifications-internals: This ... - GitHub." https://github.com/google/exposure-notifications-internals. Accessed 6 Sep. 2020.

Highlights of the paper

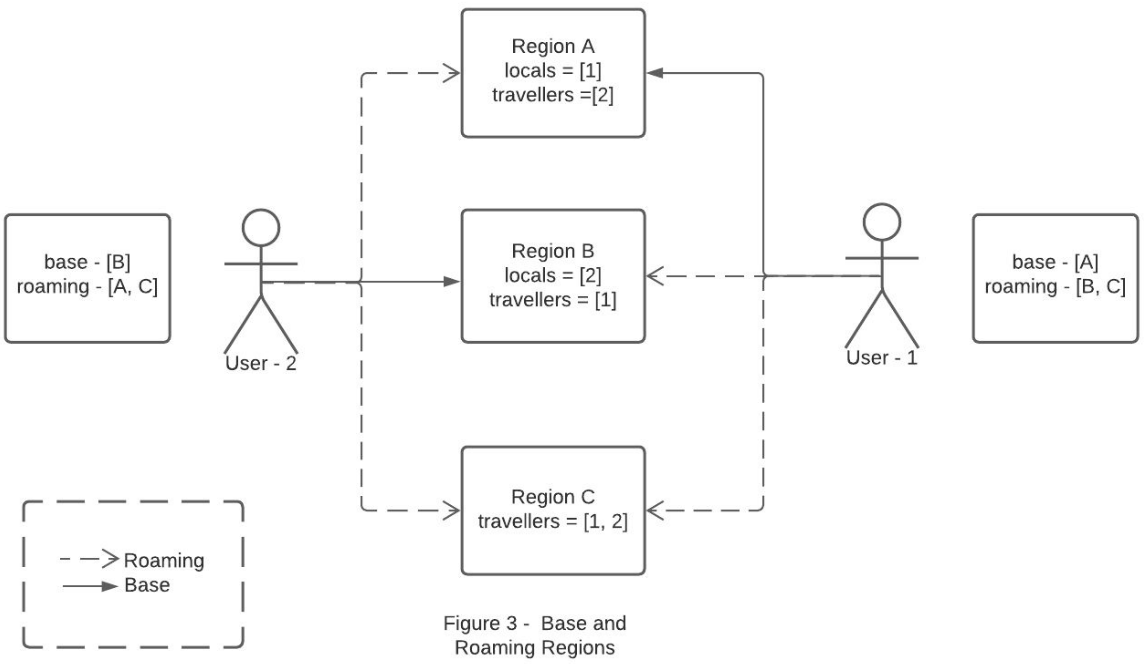


Figure 3 - Base and Roaming Regions

1. In the previous papers, We have discussed the backend bandwidth issues of a decentralized model. In this paper, Authors try to address the problem by dividing the regions to base regions and roaming regions. Users only download the keys from the regions where they visited. Based on these regions, Users are classified as local and traveller.
2. Based on the terminology of local and travellers, The paper lists 4 functional requirements based on interactions of -
    a. Local - Infected Travellers
    b. Traveller - Infected Locals
    c. Traveller - Infected Travellers
    d. Positive test of a roaming user i.e cross-country/cluster traveller
3. From the security and privacy perspective, Authors suggest signed backend servers from regions using a root certificate belonging to WHO or an organization of similar standing. Data should be deleted in 30 days.
4. To maintain information sharing among regions, Two methods were proposed for interaction between backend servers -
    a. Partial Replication - In case of cross-country traveller.
    b. Full-to-Full Replication - Common database for a single cluster of regions
5. Cross-Cluster verification of a traveller is suggested to be handled according to W3C Verifiable Credentials standard[6].

[6] "Verifiable Credentials Data Model 1.0." 19 Nov. 2019, https://www.w3.org/TR/vc-data-model/. Accessed 6 Sep. 2020.

Analysis of the Research / Use in Project

1. The paper addresses the problems of download bandwidth by only serving diagnosis keys based on regions the user stays/travels to. However, Partial replication should be preferred to full-on replication even among a single cluster like the USA or EU. Countries like India and China have a population twice or thrice of the other countries.
2. If the user is following a pull-based replication and information regarding the traveller is communicated to cross-country servers. The user's device acting as a truth value is a security risk. The paper has not addressed the security aspects of the server against attackers or improper positive results. Backend server has to authenticate the user that they have indeed visited the region - This is important to safeguard against relay-attacks.
3. The authors justify full replication among a cluster as the inter-movement among regions in quite common. During lockdowns and etc. it is not so. 45 MB for 200k daily infections is the wrong metric for bandwidth estimation. Active cases should be considered which is around 7 million as users stay positive for 14 days and they may break isolation protocols.

Region-based classification can be useful in our application to reduce the download requirements. If the server does not know the user (user can use mix servers) to fetch diagnosis keys, The app can preserve user-confidentiality while allowing the user to fetch the diagnosis keys from multiple servers. If we can somehow authenticate a user with a region's server anonymously- this approach can scale very well.

Poster Review # 2

Reelfs, Jens Helge et al. “Corona-Warn-App: Tracing the Start of the Official COVID-19 Exposure Notification App for Germany.” ArXiv abs/2008.07370 (2020): n. pag.

Why have I chosen this paper?
Since our application is based on the GAEN as protocol, I wanted to read about how well it has been adopted, the traffic metrics and the challenges faced by the German government.

Conclusion of the Research

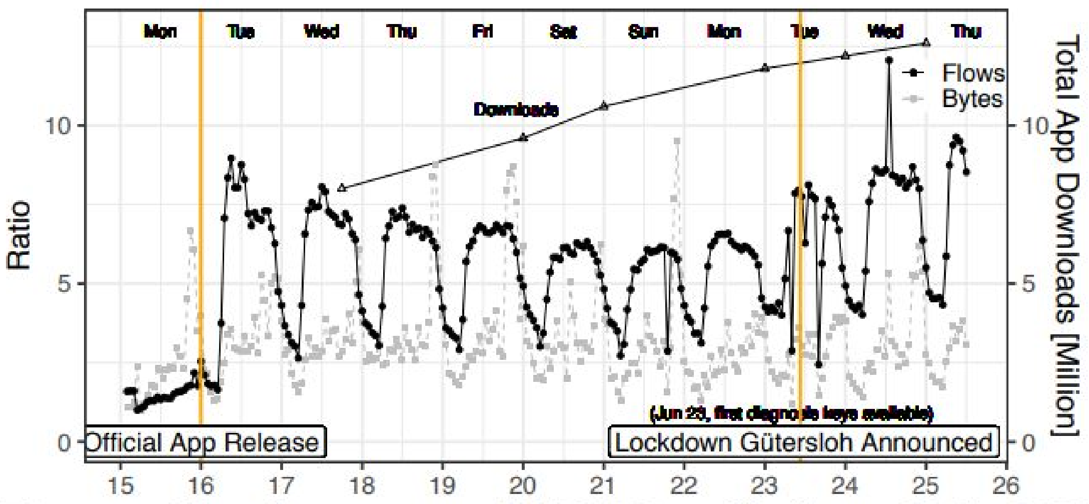


Figure 2: Hourly aggregated HTTPS traffic from CWA CDN to users normed to the minimum (left y-axis) and the total app downloads in million from Google/Apple (right y-axis).

1. There has been 7.5X increase in the CovidWarn app after release and downloaded 6.4M times after 36 hours among an 83 million population base.
2. The downloads and interests were from all districts of Germany.
3. Local outbreaks did not increase traffic in the affected regions and the traffic patterns followed nation-wide trends like lockdown as shown in the above graph.

Usage in our Project
1. CovidWarn’s client IP addresses are prefix-preserving anonymized. This is very useful for anonymising client IP addresses during calls to server. In my research for next week, I shall study it more.
2. The paper helped us in understanding the capacity planning of servers needed and the traffic to expect if we hypothetically release the app. It is very important to make sure CDN can handle the increasing workload after the release of the app and optimise

when the growth stagnates and picks up gradually from there. CDN should cater to all states of Australia. Having Point-of-Presence PoS presence in Singapore can help in reducing latencies for residents of Perth.