Mobile Computing Systems Programming

COMP90018 · Faculty of Engineering and IT (FEIT)

TrueMatch

Prof. Vassilis Kostakos · A/Prof. Jorge Goncalves

Head Tutor Mr. Shiquan Zhang

7th November 2025

Group 2 · Authors:

1659029 · Zijun Xing · zijunxing@student.unimelb.edu.au

1252032 · Shasha Wu · shashaw1@student.unimelb.edu.au

1266285 · Zherui Wang · zheruiw@student.unimelb.edu.au

1695793 · Yash Arjun Aslekar · yaslekar@student.unimelb.edu.au

1641428 · Nathan Charles Lopes · lopesn@student.unimelb.edu.au

1780512 · Karl Jorge Cleres Andreo · kcleresandre@student.unimelb.edu.au

Contents

1 App Walkthrough Video

A short walkthrough video has been prepared as an introduction to the project. It outlines the motivation behind the application, highlighting the problem it addresses and the gap it fills compared to existing solutions. The video then presents a short demonstration of the app's core functionality and provides a brief overview of the underlying algorithmic logic. The walkthrough video can be accessed via the following link:

```
https://youtu.be/wgXt_u9OYOM
```

2 High-level architecture overview

The overall architecture is illustrated in Figure 1. The system consists of three main components: the Android Phone, Firebase, and the Backend Server. The Android client collects sensor and authentication data, which is uploaded to Firebase for storage and synchronization. Firebase serves as the central data hub, managing authentication and persisting user data in Firestore. The backend server periodically retrieves this data from Firebase to perform scoring and matching operations. The resulting matching scores are then sent back to Firebase and made available to the Android client. This setup ensures a clear separation of responsibilities between data collection, storage, and computation while maintaining real-time data flow between all components.

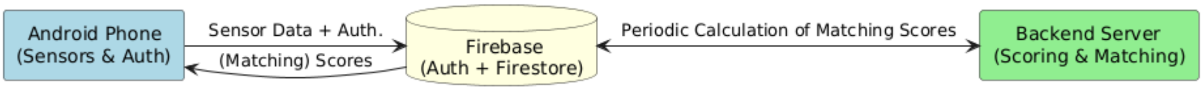


Figure 1: Logical Architecture of the Backend.

3 Prototype

A low-fidelity prototype was created in Figma to visualize the app's layout and user flow before implementation. It can be accessed at the following link:

```
https://www.figma.com/design/owWAblqoTqFcsr2LUPYHXN/TrueMatch-prototype
```

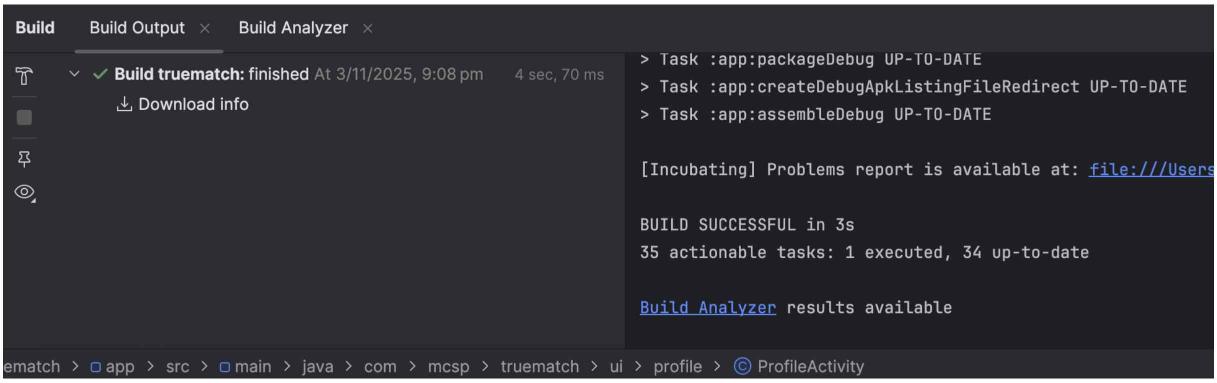
4 Project evaluation by criterion

4.1 Material

Idx	Criterion	Rating	Max. Marks
MA.1	Material - Report & Video	6	6

The report addresses all grading criteria, and the video demonstrates every key feature of the application. It is an original recording featuring real users that clearly demonstrates our unique concept and shows that the software works as intended.

Idx	Criterion	Rating	Max. Marks
MA.2	Material - Screenshot	2	2


Build
Build Output
Build Analyzer

```
Build truematch: finished At 3/11/2025, 9:08 pm    4 sec, 70 ms
    Download info
```

```
> Task :app:packageDebug UP-TO-DATE
> Task :app:createDebugApkListingFileRedirect UP-TO-DATE
> Task :app:assembleDebug UP-TO-DATE

[Incubating] Problems report is available at: file:///Users

BUILD SUCCESSFUL in 3s
35 actionable tasks: 1 executed, 34 up-to-date

Build Analyzer results available
```

ematch > app > src > main > java > com > mcsp > truematch > ui > profile > ProfileActivity

Figure 2: Screenshot of the Project Build.

Idx	Criterion	Rating	Max. Marks
MA.3	Material - Commit log	4	4

The commit log can be found in the Git-Repository under uommcsp/commit_log.txt and in the Appendix.

Idx	Criterion	Rating	Max. Marks
MA.4	Material - Itemised contributions	2	2

Table 1 presents the itemized contribution matrix for each team member, detailing the specific roles and tasks undertaken throughout the development of the software project. This matrix ensures transparent and fair allocation of work among all group members.

Contribution	Yash	Karl	Nathan	Shasha	Zherui	Zijun
Project Management		X				
Background Research	X	X	X	X	X	X
Prototype UX/UI				X	X	X
Prototype Algorithms		X				
UI/UX Design				X	X	X
Frontend Development				X	X	X
Backend Development	X	X	X			
Test Data Generation			X			
Testing & Debugging	X	X	X	X	X	X
Innovation & Technical Depth	X	X				
Report / Documentation		X				
Deployment						
Video Filming		X			X	X
Video Participation	X	X	X	X	X	X
Video Editing					X	X
Final submission preparation					X	X

Table 1: Contribution Matrix by Group Member

4.2 Implementation

Idx	Criterion	Rating	Max. Marks
IP.1	Implementation - Quality	10	10

The project code is written with a clear focus on readability, structure, and maintainability. All files follow consistent naming conventions and formatting standards, with descriptive and self-explanatory method and variable names. The project follows a MVVM architecture, which separates user interface, logic, and data handling. This structure keeps components modular, testable, and easy to extend.

Functions are concise and focused, and related code is logically grouped by responsibility. Comments are used only where necessary to clarify complex logic, while clear method names (e.g., calculateMatchingScore, updateNormalizedValues) make the code largely self-explanatory. Consistent indentation and layout across Kotlin and Python files improve overall readability and professional presentation.

Idx	**Criterion**	**Rating**	**Max. Marks**
IP.2	Implementation - Sensors	10	10

The app integrates multiple hardware and software sensors to capture diverse behavioural data. Step counts are recorded via the PedometerTrackerService, mobility is tracked using location sensors, and the SleepTrackerService infers sleep behaviour through the UsageStatsManager, showing creative use of indirect sensing. The camera supports QR-based check-ins, while app and social media usage metrics provide additional behavioural insights.

All sensors share a consistent architecture through the common BaseTrackerService, which manages periodic background execution, notifications, and wake locks. This structure, combined with proper runtime permission handling, ensures stability across different Android versions. Sensor modules integrate with the app's MVVM architecture, where their data is combined into compatibility scores that capture user behaviour in a structured way.

Idx	**Criterion**	**Rating**	**Max. Marks**
IP.3	Implementation - Connectivity	12	12

The **MainActivity** now handles a wider range of permission requests, including *location*, *usage statistics*, *activity recognition*, *camera*, *photos*, and *notifications*. Once all permissions are granted, it passes control to **LoginActivity**. When the activity is closed, it properly shuts down any active services.

The **CustomServiceManager** manages five ongoing tracking services: *location*, *social activity*, *sleep*, *pedometer*, and *app usage*. These trackers only start collecting data after a user session begins. Their data is now saved through new repositories, **UsageRepository** and **StepsRepository**, as well as an improved **FirestoreDataSource**. This setup allows the app to store location, movement, and usage information alongside each user's profile data.

Across the app interface:

- **LoginActivity** still saves the Firebase session and starts the tracking services.
- **RegisterFinalActivity** collects user media and metadata before saving it.
- **UserRepository** now includes tools for fetching and calculating match scores through the **UserDto** model.
- **DiscoveryActivity** and the updated **ChatRepository** both rely on **FirestoreDataSource** to load profiles, create date invitations, and handle responses through **DateRepository**.

Overall, these updates help maintain smooth and consistent data handling from the moment a user logs in to when they discover matches and plan meetups.

Idx	**Criterion**	**Rating**	**Max. Marks**
IP.4	Implementation - Responsiveness	4	6

The application feels responsive and smooth during everyday use, with little to no noticeable lag. The main UI thread stays lightweight, while heavier operations (such as reading sensors, processing data, or handling input/output) run in the background to keep scrolling and touch interactions fluid.

The layout automatically adjusts to different screen sizes and follows accessibility preferences for a consistent experience across devices.

Note: There might be a short delay when loading lists (like dates, chats, or likes), but once the data is ready, updates appear instantly and without lag.

Idx	**Criterion**	**Rating**	**Max. Marks**
IP.5	Implementation - Technical depth	6	6

The **BaseTrackerService** now uses a shared wake-lock system managed by **SharedWakeLockManager**. This change allows multiple tracking services to run at the same time without interfering with each other.

- **AppUsageTrackerService** observes the overall app lifecycle to measure how long the app remains in the foreground.
- **PedometerTrackerService** combines step counter sensor data with stored preferences so that step totals remain accurate even if the sensor resets.

- **SocialsTrackerService** and **SleepTrackerService** now use the shared wake-lock and the same timing system for more efficient, scheduled data updates.

On the data side, the **FirestoreDataSource** has been expanded with new methods to handle *app usage time*, *step totals*, *match scores*, and *invitation status updates.* The **UsageRepository** and **StepsRepository** provide this data to the services through Task-based APIs.

The **UserDto** model now supports fetching match scores when needed. Meanwhile, the **ChatRepository** and **DateRepository** handle Firestore transactions for creating, accepting, or declining date invitations, while keeping chat information in sync.

4.3 User Interface

Idx	**Criterion**	**Rating**	**Max. Marks**
UI.1	User Interface - Appeal	3	4

The discovery screen adopts a clean, card-style layout with a prominent *Like* floating action button and a bottom navigation for *Dates*, *Likes*, *Chats*, and *Profile*. A dedicated *Activity Scores* panel surfaces per-category signals (app usage, distance walked, mobility, sleep proxy, social usage) so key information is visible at a glance. Restraint in color, iconography and spacing keeps the surface uncluttered across light/dark themes. Overall, the application is visually appealing and pleasant due to a coherent card style, consistent spacing/typography across Discovery/Chats/Dates/QR, and dark–light theme parity.

Idx	**Criterion**	**Rating**	**Max. Marks**
UI.2	User Interface - Guidelines	5	6

The client follows Material components and platform navigation patterns. Long-running sensing and Firestore I/O run off the main thread to avoid jank; screens use repositories and (in the auth flow) an MVVM `AuthViewModel` for predictable state. Touch targets, system themes and dynamic type are respected; standard bars/sheets are used for navigation and feedback. It explicitly follows Android UI guidelines (Material components, 48 dp targets, contrast and dynamic type), aligning the interface with the OS conventions.

Idx	**Criterion**	**Rating**	**Max. Marks**
UI.3	User Interface - Flow	6	6

Core journeys are short and free of dead-ends:

- **Onboarding → Discovery:** Email/password login leads directly to discovery.
- **Discovery → Conversation/Lists:** Bottom navigation opens *Chats*, *Likes*, or *Dates*; the Like FAB records interest without extra steps.
- **Conversation → Planning:** From a chat thread, the *Schedule* header action opens the date screen; time/place are prefilled when available and can be edited before sending.
- **Planning → Tracking:** The Dates screen aggregates invitations and shows statuses (*Pending*, *Accepted*; others are filtered/expired).

It is clear and easy to access/find/activate features: primary actions are one tap away (Like FAB, chat *Schedule*); secondary areas are always reachable via bottom navigation.

Idx	**Criterion**	**Rating**	**Max. Marks**
UI.4	User Interface - Language	4	4

Microcopy uses plain terms and neutral tone. Status labels like *Pending* and *Accepted* are unambiguous (Declined/Expired are suppressed on the list view). The *Activity Scores* section uses everyday category names (*App usage*, *Distance*, *Mobility*, *Sleep*, *Social*) to keep explanations readable. Labels and messages are meaningful and appropriate, avoiding jargon while making system state explicit.

Idx	**Criterion**	**Rating**	**Max. Marks**
UI.5	User Interface - Reactiveness	5	6

Foreground services handle pedometer, usage stats, location, and sleep-proxy sampling on intervals; Firestore access uses caching/writes and avoids blocking the UI thread. Lightweight loading/overlay patterns (e.g., non-modal progress, QR overlay during scanning) preserve interactivity even on variable networks and sensor schedules. The interface dynamically reacts to sensors/external data: Activity Scores reflect live sensor uploads, date statuses update in real time via Firestore listeners, and the QR view changes state immediately on successful scan.

4.4 Innovation

Idx	**Criterion**	**Rating**	**Max. Marks**
IN.1	Innovation - Novelty	3	3

Matching and discovery are grounded in passively sensed behaviour rather than static self-description. The app collects steps, app-usage, coarse location, and a sleep proxy, normalises them into `NormalizedScores`, and surfaces per-category signals via an *Activity Scores* panel. Behaviour-grounded discovery and visible per-category scores are novel compared with profile-only matching.

Idx	**Criterion**	**Rating**	**Max. Marks**
IN.2	Innovation - Surprise	3	3

From the chat thread, one-tap scheduling opens a dedicated screen that pre-fills time/place when available and creates a shareable date invite; an optional QR check-in validates attendance with a lightweight overlay. chat→schedule prefill and in-person QR check-in are unexpected yet practical extensions to standard dating apps.

Idx	**Criterion**	**Rating**	**Max. Marks**
IN.3	Innovation - Tech Knowledge	4	4

Android services implement pedometer, usage stats, location, and a sleep-proxy; Firebase Auth/Firestore provide identity and real-time data, with repositories encapsulating access. `DateRepository` manages invite lifecycle (pending, accept, expire); `ChatRepository` streams messages; MVVM is applied in the auth flow with `AuthViewModel`. This demonstrates integration of Computing and Information Systems knowledge across mobile, cloud back-end, data modelling, and UI architecture.

Idx	**Criterion**	**Rating**	**Max. Marks**
IN.4	Innovation - Cross-Disciplinary	3	3

The work blends mobile sensing and HCI (explainability through category scores, consentful flows) with lightweight data modelling and real-time cloud sync; server notebooks outline scoring/similarity calculations. It draws from ubiquitous computing, UX research, and applied data science and applies them coherently.

Idx	**Criterion**	**Rating**	**Max. Marks**
IN.5	Innovation - Impact	3	3

Scheduling shortcuts and clear status flows reduce effort to organise meetings and lower drop-off between chat and first meeting. Visible behaviour categories set clearer expectations than purely aesthetic profiles; QR check-in provides a minimal, privacyrespecting confirmation of attendance. The application has clear potential to improve users' coordination tasks and meeting outcomes.

Test users' account

- **Luke Harris (male)**
  Email: `luke@mail.com` PW: `luke1234`
- **Mia Patel (female)**
  Email: `mia@mail.com` PW: `mia12345`
- **Jack Nguyen (male)**
  Email: `jack@mail.com` PW: `jack1234`
- **Chloe Martin (female)**
  Email: `chloe@mail.com` PW: `chloe123`
- **Ethan Walker (male)**
  Email: `ethan@mail.com` PW: `ethan123`
- **Olivia Papadopoulos (female)**
  Email: `olivia@mail.com` PW: `olivia12`

Usage of Artificial Intelligence (AI)

This work was created using standard writing and coding processes, with AI-based tool assistance. The following AI-based tools were used:

- ChatGPT - OpenAI Inc., `https://chatgpt.com/`
- Deepseek - Hangzhou DeepSeek AI Co. Ltd., `https://deepseek.ai/`
- DeepL - DeepL S.E., `https://deepl.com/`
- Gemini - Google Inc., `https://google.com/`
- LanguageTool - LanguageTooler GmbH, `https://languagetool.org/`
- IntelliJ Code Completion - JetBrains s.r.o., `https://www.jetbrains.com/help/idea/auto-completing-code.html`
- Claude - Anthropic PBC, `https://claude.ai/`

A.1 Evaluation Summary

Idx	**Criterion**	**Rating**	**Max. Marks**
MA.1	Material - Report & Video	6	6
MA.2	Material - Screenshot	2	2
MA.3	Material - Commit log	4	4
MA.4	Material - Itemised contributions	2	2
IP.1	Implementation - Quality	10	10
IP.2	Implementation - Sensors	10	10
IP.3	Implementation - Connectivity	12	12
IP.4	Implementation - Responsiveness	4	6
IP.5	Implementation - Technical depth	6	6
UI.1	User Interface - Appeal	3	4
UI.2	User Interface - Guidelines	5	6
UI.3	User Interface - Flow	6	6
UI.4	User Interface - Language	4	4
UI.5	User Interface - Reactiveness	5	6
IN.1	Innovation - Novelty	3	3
IN.2	Innovation - Surprise	3	3
IN.3	Innovation - Tech Knowledge	4	4
IN.4	Innovation - Cross-Disciplinary	3	3
IN.5	Innovation - Impact	3	3
$\sum$		**95**	100

Firebase Firestore Datamodel

Listing 1: Logical Firestore Data Model

```
firebase/
|-- chats/
|   ‘-- {chatId}
|       |-- participants: [userId, userId]
|       |-- lastTimestamp: ISO-8601
|       |-- unreadBy: userId or null
|       ‘-- lastMessage: string
|
|-- dates/
|   ‘-- {dateId}
|       |-- timeIso: string
|       |-- location: string
|       |-- status: "pending" | "declined" | "expired" | "accepted" | "checkedIn"
|       |-- inviterId: userId
|       |-- inviteeId: userId
|       |-- updatedAt: ISO-8601
|       |-- createdAt: ISO-8601
|       ‘-- chatId: {chatId}
|
|-- likes/
|   ‘-- {likeId}
|       |-- likerId: userId
|       |-- likedUserId: userId
|       ‘-- timestamp: ISO-8601
|
|-- matches/
|   ‘-- {matchId}
|       |-- participants: [userId, userId]
|       |-- isActive: boolean
|       ‘-- timestamp: ISO-8601
|
‘-- users/
    ‘-- {userId}
        |-- (...) collections with raw user data (...)
        |-- userID: userId
        |-- email: string
        |-- lastName: string
        |-- biography: string
        |-- gender: string
        |-- profilePictureBase64: base64
        ‘-- normalized_scores:
            |-- appUsage: number
            |-- dateActivity: number
            |-- distanceWalked: number
            |-- mobility: number
            |-- sleep: number
            |-- socialMediaUsage: number
            ‘-- updatedAt: timestamp
```

Listing 1: Logical Firestore Data Model

Git Commit Log

```
651f6b7 | cleresk | 2025-11-03 | Merge remote-tracking branch 'origin/main'
4253d90 | cleresk | 2025-11-03 | Added caching techniques!
e23b191 | cleresk | 2025-11-03 | Revise backend run instructions in README
5612257 | Nathan Lopes | 2025-10-31 | Update README.md
e0ce271 | Nathan Lopes | 2025-10-31 | Create README.md
be52885 | Nathan Lopes | 2025-10-31 | Add files via upload
cf1e43e | Nathan Lopes | 2025-10-31 | Delete truematch_server
63b068e | Nathan Lopes | 2025-10-31 | Create truematch_server
fb5e7c7 | Nathan Lopes | 2025-10-31 | Delete truematch_server/deleteme.txt
2cabada | cleresk | 2025-10-30 | Added README.md
7806695 | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/settings.gradle.kts
80b09bb | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/local.properties
4c77398 | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/gradlew.bat
7ff0176 | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/gradlew
f276aa0 | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/gradle.properties
6f6cefc | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/build.gradle.kts
79b759f | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/gradle directory
3675aff | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/app directory
a600ed8 | Yash Aslekar | 2025-10-30 | QR code logic for later integration into the app
581ab9f | Yash Aslekar | 2025-10-30 | Delete qr_scan_logic/app
4a604c3 | Yash Aslekar | 2025-10-30 | Create app
9693a80 | Yash Aslekar | 2025-10-30 | QR code logic for later integration into the app
```

```
05787dc | cleresk | 2025-10-29 | Added yash and nathans folders
0808285 | cleresk | 2025-10-29 | Refactoring and cleaning the project code!
413d42a | cleresk | 2025-10-29 | Added Filter function
ddbf2ae | cleresk | 2025-10-29 | Changed minor errors
c992763 | Lachie-Xing | 2025-10-25 | The profile also distinguishes between null values and 0
96fb9ec | Lachie-Xing | 2025-10-25 | Modified the visual representation of null values
51bbfe8 | Lachie-Xing | 2025-10-25 | Added the ability to delete languages
c0e8475 | Lachie-Xing | 2025-10-25 | Differentiate between zero and null values
4cba097 | cleresk | 2025-10-25 | zw
0b6df2a | Lachie-Xing | 2025-10-25 | Merge remote-tracking branch 'origin/main'
99fa9d3 | Lachie-Xing | 2025-10-25 | change user_details page
0e68335 | zheruiiii | 2025-10-24 | Merge remote-tracking branch 'origin/main'
fed7a40 | zheruiiii | 2025-10-24 | Discovery page displays scores of each user
1664559 | Lachie-Xing | 2025-10-24 | Debugging
944caf5 | Lachie-Xing | 2025-10-24 | Debugging
0cc3590 | Lisa-student | 2025-10-23 |  Implement profile page
a12cb99 | cleresk | 2025-10-23 | refactored project
8700473 | cleresk | 2025-10-23 | minor changes
2f3ec19 | Lisa-student | 2025-10-22 |  implement Profile page (shown the activity scores)
a1163a8 | Lisa-student | 2025-10-22 | Add ProfileEditDescription page
4262b52 | Lisa-student | 2025-10-22 | Add ProfileEditBirthday page
974819f | Lisa-student | 2025-10-22 |  Add ProfileEditLanguage page, with new language list
3bfe5cb | Lisa-student | 2025-10-22 | Add ProfileEditEmail page, it can also edit password
```

```
d28b529 | Lisa-student | 2025-10-22 | Add ProfileEditPhoto page
4026080 | Lisa-student | 2025-10-22 |  Add ProfileEditGender page
9b7e76f | Lisa-student | 2025-10-22 |  Add ProfileEditName page
5bcb7b3 | Lisa-student | 2025-10-22 | Add the ProfilePage, implement the ProfileReviewPage
d9f9d02 | Lisa-student | 2025-10-22 | managed to retrieve prof pic from firebase
920ffd4 | zheruiiii | 2025-10-22 | QR code check in logic
3f398d2 | zheruiiii | 2025-10-22 | Date schedule status (Pending/Accepted) can be updated by date receiver's response
3c23216 | zheruiiii | 2025-10-21 | DateSchedule page, date invitation sender and receiver different interface
dca5888 | zheruiiii | 2025-10-19 | New date invitation overwrites the old invitation
e7df18c | cleresk | 2025-10-16 | example on how to use backend to get the matching and individual scores
50d62f7 | cleresk | 2025-10-16 | added missing services
743c47e | zheruiiii | 2025-10-16 | Debugging
4c84df6 | zheruiiii | 2025-10-16 | can modify the location and time of the date invitation that has been sent.
a9d552e | zheruiiii | 2025-10-16 | Merge remote-tracking branch 'origin/main'
e060199 | zheruiiii | 2025-10-16 | Different interface for Date senter and Date receiver
af22ca4 | Lisa-student | 2025-10-16 | Merge branch 'main' of https://github.com/cleresk/uommcsp
716bbb1 | Lisa-student | 2025-10-16 | Enhance ProfileActivity with gradient background and improved UI elements
dd9f3fb | zheruiiii | 2025-10-16 | Finalize chat UI features and implement send a date function
4e8b303 | Lisa-student | 2025-10-14 | Update profile page with enhanced UI and functionality
c7887a7 | Lachie-Xing | 2025-10-13 | Debugging
f3a84d6 | Lachie-Xing | 2025-10-13 | Changed the logic of user registration account in the front-end code
```

```
05a6b6d | Lachie-Xing | 2025-10-13 | Implemented likelistpage and its related functions; modified some UI views in chatlistpage
593801b | Lachie-Xing | 2025-10-13 | Updated the content of the user details page to display the user's distance information, social software usage and sleep status
b2f15c8 | cleresk | 2025-10-11 | LocationTrackerService, SleepTrackerService and SocialMediaTrackerService implemented
f54517d | Lachie-Xing | 2025-10-09 | Debugging
091e57c | Lachie-Xing | 2025-10-09 | Debugging
efe0228 | zheruiiii | 2025-10-07 | push missing xml file
536c8f6 | zheruiiii | 2025-10-04 | Debugging
2a55130 | zheruiiii | 2025-10-04 | Fix navigation bar page redirection and layout in Discovery
2ae235d | zheruiiii | 2025-10-03 | Enable bottom Navigation bar navigates to profileActivity page
7c33609 | Lachie-Xing | 2025-09-30 | Fixed minor code errors
68e3ea7 | Lachie-Xing | 2025-09-30 | fix(layout): correct RecyclerView orientation attribute in activity_chat_list.xml to resolve resource compilation error; fix(layout): add missing xmlns:app in item_like_user.xml and correct RecyclerView orientation in activity_chat_list.xml
8e4507f | Lachie-Xing | 2025-09-30 | add like list and bottom navigation to chat page
a253e47 | Lachie-Xing | 2025-09-28 | Create a usable discovery page
904c961 | Lachie-Xing | 2025-09-25 | test change
a8167e5 | cleresk | 2025-09-24 | Started doing SocialMediaUsageService
1fe35f1 | cleresk | 2025-09-24 | Implemented advanced chat functions! Finished chat! Did not debug though!
74d18d4 | cleresk | 2025-09-24 | Implemented basic chat function
dc5b5e0 | Lachie-Xing | 2025-09-22 | Implemented the register function
98226c4 | Lachie-Xing | 2025-09-22 | Modify the register pages
e74cbd3 | Lachie-Xing | 2025-09-22 | Improved the user registration screen
```

```
38afa4d | zheruiiii | 2025-09-18 | Merge remote-tracking branch 'origin/main'
831f129 | zheruiiii | 2025-09-18 | LoginActivity create account navigates to This is me
62229cd | cleresk | 2025-09-18 | Added locaton serice
d79fd9d | Lisa-student | 2025-09-18 |  implement useractrivitydetail more similar to figma
3bca412 | Lisa-student | 2025-09-18 | implement UserDetailActivity
250da2b | Lachie-Xing | 2025-09-15 | Add new drawable resources for login/signup pages
83180df | Lachie-Xing | 2025-09-15 | Modified the splash screen when launching the program; Modified the UI of the login interface; Added new button colors and icons
e330886 | cleresk | 2025-09-13 | Added fictional user to firebase. Username: k.cleres@icloud.com, password: poppers1234
4503478 | zheruiiii | 2025-09-11 | Page_for_ThisIsMe_Gender_Photo
c22e427 | zheruiiii | 2025-09-11 | Page_for_ThisIsMe_Gender_Photo
9b2a77a | zheruiiii | 2025-09-11 | Page_for_ThisIsMe_Gender_Photo
de62678 | Lisa-student | 2025-09-11 | fix wrong node name in MainActivity.java
58c63f6 | zheruiiii | 2025-09-11 | a
989df55 | cleresk | 2025-09-11 | Stellas changes
8e49248 | cleresk | 2025-09-11 | gitignore
37d2352 | cleresk | 2025-09-08 | Added template for ui and utils
95cdf75 | cleresk | 2025-09-08 | Adde firebase support for createUser()
9acc87a | cleresk | 2025-09-08 | Adde firebase support for createUser()
4dcd0ee | cleresk | 2025-09-04 | Added Firebase dependencies
2e853a3 | cleresk | 2025-09-04 | Implemented Firebase Connection
35c3372 | cleresk | 2025-09-04 | Working empty project
```

Acronyms

AI Artificial Intelligence.

MVVM Model–View–ViewModel.