

# HD Project Proposal: Chronos

Name: Ekrar Efaz | Student ID: 103494172  
Tutor Name : Dr. Naveed Ali  
Tutorial: 10:30 AT325

## 1. Problem Description

Chronos Society is a social organization based in Hawthorn. The organization aims to have serious timepiece enthusiast as its members. Currently, the board of directors has decided to take on a digital approach to their love for timepieces. They want to publish an app that would keep track of a member's timepieces and his future endeavours in the world of watches along with be-friending other members on the platform to share their interests and collections. The objective of the project is to develop an internal software that allows members to save their collection and digitally interact with other members within the organization. The software is designed with a graphical interface feature and a database to store all data securely.

## 2. Scope

The project aims to digitize a user's watch collection and present an interface for all users to share their collection and send request for exchanges.

The user can add timepieces to his collection also remove it. The user can add photo of his timepiece and the collection is displayed in the software in a listed format. The data is stored in a database that is connected to the software.

The application doesn't handle financial transaction in case a trade or exchange has been made nor does it keep track of the exchanges or trades. Trades are solely the user's responsibility. The application also doesn't host any pre-loaded database of timepiece models which a user might wish to navigate.

## 3. Stakeholders

The frontline stakeholder is the customer i.e. The Chronos Society. Their interest is what forms the final product. The most important factor is to satisfy the directors of The Chronos Society. They hold the most power and influence in deciding the outcome of the product we are working on. Their requests and requirements for the interface and functionality of the software are given the highest priority.

Another important stakeholder for the product is our team. The team translates the companies requests in code lines and they steer the direction of the application's outcome and have power to influence the clients based on ease of implementation and stronger security. We as stakeholders, are responsible for completing the tasks in assigned time with near perfect functionality.

## 4. Deliverables and Schedule

We plan to provide the client with a working prototype in Sprint 1. They will be provided with a user manual to test the functionalities of the prototype. With their approval of the prototype, we work to implement additional tweaks for the application and address the performance concerns. After completion of the development phase, we will provide the client team with Binaries, Source Code, Software Testing Results and a User Manual as a complete product.

## 5. Release Schedule of Product Backlog Items

Index	Item	Dependencies	Business Value (1 least - 10 most)	Release Schedule (Sprint)
F1	Add New Watch	NA	8	Sprint 1
F2	Add User	NA	9	Sprint 1
F3	Remove User	F2	9	Sprint 1
F4	Capture Watch Photo	F1	5	Sprint 1
F5	User Authentication	F2	6	Sprint 1
F6	Display Watch Collection	F1, F2	8	Sprint 1
F7	Add Friends	NA	6	Sprint 2
F8	Remove Watch	F1	7	Sprint 2
F9	Send Messages	F8	5	Sprint 2
F10	Search in Collection	F1, F7	6	Sprint 2
F11	Toggle Light/Dark mode	NA	2	Sprint 2
F12	Daily Promotional Mails from Organization	NA	7	Sprint 2
F13	Report Daily user logon	F1	8	Sprint 2

## 6. Solution Domain

### a. Problem Domain

- Social Organization
- Personal Collection
- Luxury Retail Tracker
- Data Management

### b. Solution Domain

- Data Management System
- Live Online Database
- Mobile Application
- Messaging System

### c. Knowledge in Related Field

- Have used and Developed applications for personal tracking and data management
- Not experienced in Luxury Retail Items.

### d. Knowledge in Related Programming Skills

- We are team of experienced android developers who worked on various projects developing multifarious applications for different use cases.
- We also have members with exceptional database management skills.

### e. Solution Writeup

We will be working on an Android Application based on a survey done before the initiation of the project indicated that 93% of the Chronos members use Android device. Our plan to create a web-based application was not taken well by the Client because most of the users of the application will be in a middle age range and they won't be comfortable using a web application. The client also wanted portability which an android application provides the best solution for.

The application will be used by different members of the Chronos Society and should be on portable device. So, we want to create an android application that the users will install on their preferred android device and use the login credentials provided by the Chronos sysadmins. The data will be hosted on a cloud database (Firebase) to ensure maximum uptime and easier access.

The android application will be divided into several activities starting with a login screen. The login authentication will be conducted by Firebase Authentication services.

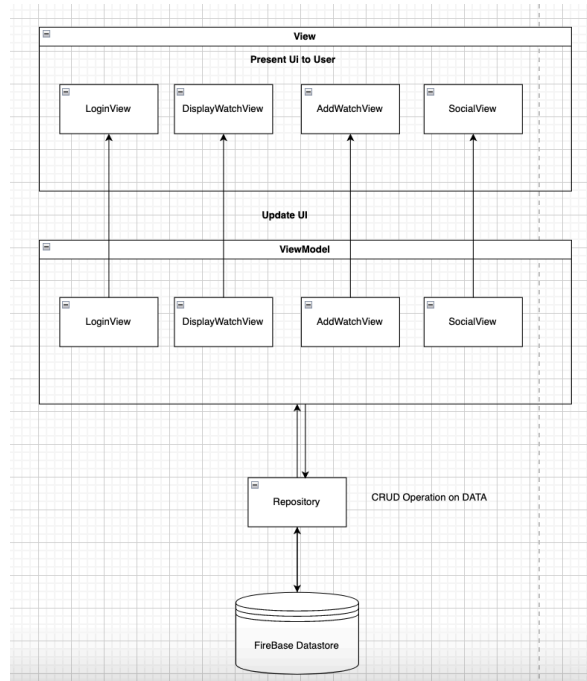
The homepage will display two clickable options, one would lead the user to his collections and wish lists and the other would direct him to a social feed that would take data from firebase live-data server and display it to the user.

The collection page would be displayed via Android Recycler View and the list would handle click events to make changes to each item.

## 7. Solution Architecture

The architecture we are going for is the MVVM (Model-View-ViewModel) architecture. It is one of the most popular architectures in mobile development. It is popular because of its modularity. MVVM separates the functionalities of UI/UX, Business Logic and Datastore into 3 parts. View Displays the UI/UX to the user and has no logical backend functionalities. The ViewModel is called when a click action happens on the view and the ViewModel is the middleman between the Database and the View. The ViewModel updates the View after updating the database via the repository.

Every View has a ViewModel which is responsible for handling its updates and click events. The repository is the communicator between a datastore and the ViewModel. In our case Firebase API is the repository that communicates with the ViewModels.



## 8. Criteria Formulation

Factors	Justifications	Importance Scale 1 - 5
Business Value	The business value should be highly considered since the business's stand personally and its worth is important as it bring impacts towards the employee and the client itself.	5
Feature Dependency	Feature dependency should be considered because some backlog items are dependent on each other. Therefore, the order of the backlog items should be based on their dependencies.	4
Risk Involved	Risk Involved should be considered because some tasks that have high risks might take a lot of time to finish. If we consider "Risk Involved" as the criteria to evaluate the backlog items, we can ensure that the high-risk tasks always have backup time in case anything goes out of track.	3

### Justification for Backlog Prioritization

- The business value factor has the highest emphasize because the program must ensure it can provide the required features for the stakeholders. Therefore, these features must be considered as the first item in the 'To-do list', so that we can demonstrate to our stakeholders.
- Feature dependency is also one of the important things to do as some of the backlog items are dependent to each other. Dependencies can bring benefit towards the project by creating and managing lead and lag times. Lead time is duration of time which an activity can be accelerated or advanced in relation to the predecessor task, while the lag time is duration of time which a successor task is delayed in relation to the predecessor activity. This functionality is efficient when reschedule a task, all dependent tasks are automatically rescheduled rather than do update schedule for single task manually.

- Risk Involved is high factor as well as we must identify and analyse threats to an organization's capital. Everyone must manage risk and makes the jobs safer to enables the successful project.

As per the justifications mentioned, the following backlog items were chosen for sprint 1 ordered by their working sequence.

#### I. Add User

This Item is the functioning backbone feature of the application. Since the application is user based the addition of a user holds the highest priority.

#### II. Add Watch

Adding a watch is also a primary feature of the app requested by the client. The applications aim is to display and track watch collections. So, adding a watch is one of the highest priorities we must deliver.

#### III. Remove Watch

Remove watch does exactly what the name suggests. It removes watches from a collection added by the user. Deletion always proves to be a fundamental feature in every situation. The user might not have a watch in his collection anymore and would want to remove it from app database as well. Thus, deletion is on the higher end of the priority scale.

#### IV. User Authentication

All the user accounts are determined by the client organization to make sure only the elite members are given the authorization to use the application. So, authentication of users based on password and username is on higher priority

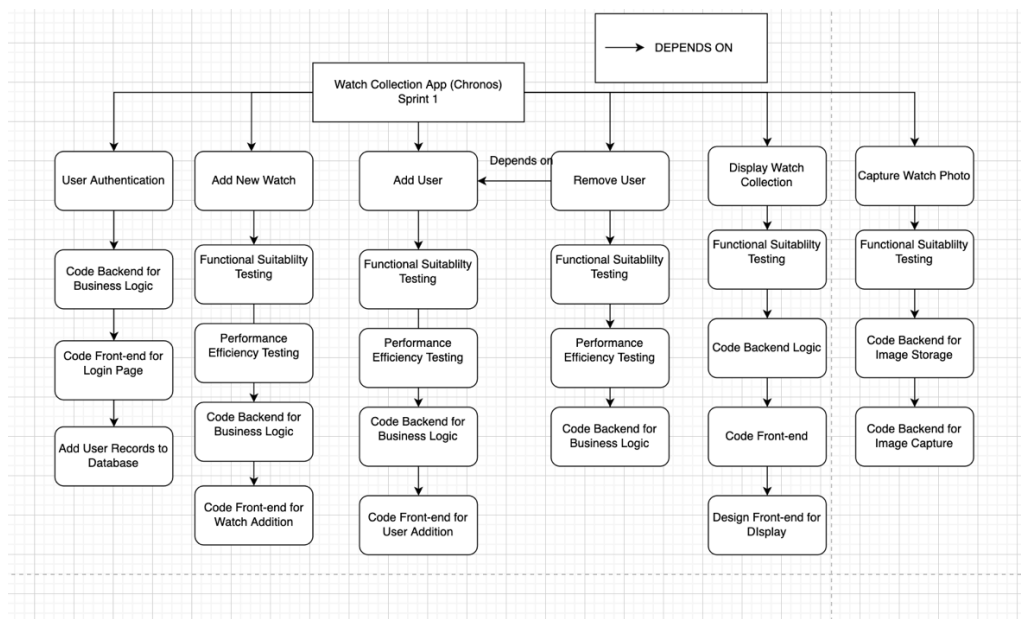
#### V. Display Watch Collection

The user would want to see all his collections in a single page. The display of a user's collection is one of the key features of the application. That's why it is put on high priority.

#### VI. Capture Watch photo

Capturing and add a watch photo to the collection database is a nice feature of the application agreed upon by the client upon proposal. This item depends on the higher priority items such as add user and add watch.

### 9. WBS



## 10. Quality Management

### ***Backlog Item: Add New Watch***

- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional completeness
    - Quality metric/ Test condition: Total tests failures
    - Threshold: Less than 8% of total test cases have failed results.
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.
- Quality characteristic: Performance Efficiency
  - Sub-characteristic: Time Behaviour
    - Quality metric/ Test condition: Time for executing one task
    - Threshold: Executing a task takes less than 2 seconds.

### ***Backlog Item: Add User***

- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional completeness
    - Quality metric/ Test condition: Total tests failures
    - Threshold: Less than 8% of total test cases have failed results.
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.

### ***Backlog Item: Remove User***

- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional completeness
    - Quality metric/ Test condition: Total tests failures
    - Threshold: Less than 8% of total test cases have failed results.
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.
- Quality characteristic: Security
  - Sub-characteristic: Authenticity
    - Quality Metric: Number of successful delete attempts without authentication
    - Threshold: No. of successful attempts should be < 3%

### ***Backlog Item: Capture Watch Photo***

- Quality characteristic: Usability
  - Sub-characteristic: Operability
    - Quality metric/ Test condition: Mean score evaluated by the users (on a scale of 10)
    - Threshold: The mean score is greater than 8
- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional completeness
    - Quality metric/ Test condition: Total tests failures
    - Threshold: Less than 8% of total test cases have failed results.
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.

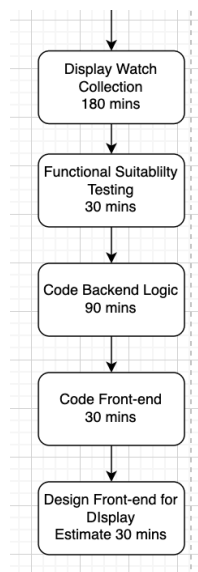
### ***Backlog Item: User Authentication***

- Quality characteristic: Reliability
  - Sub-characteristic: Fault Tolerance
    - Quality metric: No. of false positives during authentication
    - Threshold: For total number of logins the false positives should be < 2%.
- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.

**Backlog Item: Display Watch Collection**

- Quality characteristic: Functional suitability
  - Sub-characteristic: Functional completeness
    - Quality metric/ Test condition: Total tests failures
    - Threshold: Less than 8% of total test cases have failed results.
  - Sub-characteristic: Functional correctness
    - Quality metric/ Test condition: Correct tasks done
    - Threshold: 100% of the total tasks are done correctly.
- Quality characteristic: Usability
  - Sub-characteristic: Operability
    - Quality metric/ Test condition: Mean score evaluated by the users (on a scale of 10)
    - Threshold: The mean score is greater than 8
- Quality characteristic: Security
  - Sub-characteristic: Confidentiality
    - Quality metric/ Test condition: Number of valid accesses to the data
    - Threshold: No invalid access occurs.
- Quality characteristic: Portability
  - Sub-characteristic: Adaptability
    - Quality metric/ Test condition: The ability to display the data in the mobile's screen size.
    - Threshold: All data is displayed in the mobile.

**11. Effort Estimation**



### *Displaying Watch Collection (Estimate 180 mins)*

The display of watch collection involved designing the front-end followed by coding the frontend. The data was taken from the cloud database which were then displayed in a list form. The list was refactored to enhance performance because sometimes less powerful android devices lagged on list scroll. Finally, some tests were run to ensure the application was functioning as it was intended to. The whole process took 180 mins in total.

### *Design Front-end for Display (30 minutes)*

#### a. Estimation by Analogy

I have designed a lot of Android Application frontend and each design required an estimate of 10 minutes on sketch and 20 minutes to develop a prototype to present to the client. For displaying watch collection to a user my design was a list form that showed all the watch data along with an added photo.

#### Previous Experience

- Design ListView frontend for a Book tracking android app.
- Time Required: 25 minutes

#### Current Task

- Design RecyclerView frontend for a watch collection app.
- Estimated Time: 30 minutes

### *Code Frontend for Display Collection (30 minutes)*

#### a. Estimation by Analogy

I have coded and designed android frontend for a few different applications. I can easily work with android studio and XML to code a frontend that was prototyped for the client's approval.

#### Previous Experience:

- Working on Android UI/UX using Android Studio
- Code a *ListView* frontend for Book Tracking app.
- Time required: 30 minutes

#### Current Task

- Code a "*RecyclerView*" frontend to display a watch collection read from database
- Estimated Time: 30 minutes

### *Code Backend for business logic (90 minutes)*

#### a. Estimation by Analogy

I am very familiar with RecyclerView and how they are implemented for performance. I have done several RecyclerView code for different projects and even to help my friends with their android application to achieve better performance. The current task is a high risk and high business value backlog item.

#### Previous Experience

- Coding backend for RecyclerViews and grabbing the data from a database
- Reading CSV data from a database to display Olympic Stats in a RecyclerView
- Time Required ~80 minutes

#### Current Task

- Read JSON data from a cloud database
- Display the data in a RecyclerView
- Estimate 90 minutes

#### *Functionality Testing*

##### a. Estimation by Comparison

we have done functionality testing before in other backlog items and compared to functionality testing for Displaying Member data this is also similar. The effort required for testing is same.

#### *Comparison Table*

Add User	Display watch Collection
Functional Suitability Testing	Functional Suitability Testing
Time Required: 33 minutes	Time Estimate: 30 minutes