## **Chapter 3 Project Activities**

This chapter provides a comprehensive overview of the project activities, organized into weekly sprints that guide the team's efforts throughout the duration of the project.

### **Sprint-Schedule Overview**

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Dates** | **Demo / review date** |
| **Sprint 1** | 12 May – 25 May 2025 | Fri 23 May |
| **Sprint 2** | 26 May – 8 Jun 2025 | Fri 6 Jun |
| **Sprint 3** | 9 Jun – 22 Jun 2025 | Fri 20 Jun |
| **Sprint 4** | 23 Jun – 6 Jul 2025 | Thu 3 Jul (final) |

### **Sprint 1: Planning & Foundations**

### **Objectives:**

* Define the app concept and architecture.
* Prepare planning, documentation, and tools.
* Begin security integration via Threat Modeling.

**Activities:**

* Brainstorm and finalize app concept with team.
* Define main and sub-features of the application.
* Create product backlog and user stories in Scrum tool (GitLab).
* Setup GitLab repository and branching strategy.
* Create initial README, code of conduct, license file.
* Define security goals.
* Create **Threat Model** (STRIDE or OWASP).
* Create Planning documents:
  + Application concept document
  + Plan of Action (PoA)
  + Secure SDLC strategy (what secure practices will be used in each phase)
  + List of external APIs and OpenData sources.

**Deliverables:**

* App concept document
* Threat Modeling
* Planning document
* Initial Git repo with README

### **Sprint 2: Secure Development – Backend & Frontend (Week 3–4)**

**Objectives:**

* Develop core backend and frontend features.
* Integrate secure coding principles from OWASP.
* Apply authentication method.

**Activities:**

* Design app screens (wireframes/mockups).
* Implement Kotlin Android app architecture (MVVM preferred).
* Build UI screens (login, dashboard, stats view).
* Set up secure authentication (Firebase Auth, OAuth2, etc.).
* Use **OpenData APIs** and **2 external APIs** (e.g., game stats).
* Apply secure coding practices:
  + Input validation
  + Secure storage (SharedPreferences with encryption)
  + Avoiding hardcoded secrets
* Track all commits in Git.

**Deliverables:**

* Functional app skeleton
* Authentication setup
* API Integration
* Secure programming implementation evidence

### **Sprint 3: Cryptography, Secure Testing & Code Review (Week 5–6)**

**Objectives:**

* Apply **Cryptography** and secure testing practices.
* Conduct **Code Review** and improve code quality.

**Activities:**

* Integrate cryptographic methods.
* Implement secure local storage using encryption.
* Perform unit tests, UI tests.
* Peer code review sessions (mandatory).
* Document security measures taken.
* Apply **static code analysis** and linters (e.g. Android Lint).
* Draft report chapters:
  + Secure Programming
  + Cryptography
  + Secure SDLC
  + Code Review
  + Testing

**Deliverables:**

* Secure crypto implementation
* Testing & code review documentation (mandatory appendix)
* Updated Git history with tags/releases
* Draft report with technical chapters

### **Sprint 4: Final Integration, CI/CD, Report & Presentation (Week 7–8)**

**Objectives:**

* Finalize application, set up CI/CD.
* Complete documentation and report.
* Prepare and deliver the presentation.

**Activities:**

* Set up CI/CD pipeline in GitLab:
  + Build trigger
  + Linter
  + Static code analysis
  + Test runner
* Finalize all features and UI.
* Conduct usability testing if time allows.
* Compile final report (12,000–16,000 words) with:
  + Summary (in Dutch or English depending on program)
  + Introduction
  + Planning/Methodology
  + Secure programming application
  + Cryptography
  + Secure SDLC
  + Threat modeling
  + Code review
  + Conclusion & recommendations
  + APA-referenced sources
* Deliver group presentation (5–10 minutes per member).
* Peer-assessment form for group contribution.

**Deliverables:**

* Working, secure app in GitLab with proper CI/CD
* Final report (PDF with appendices)
* Final presentation (PPT/video)
* Peer assessment (point distribution)

## Chapter 8: Planning

Additional elaboration concerning the planning will be done via the Gantt chart, which will be attached in addition to this document. The Gantt chart is consistent with the project activities stated in chapter 3.  For a more detailed and clearer view check the file name VGC Gantt in the documents folder.

