**Attendance Management System** – A system to track and manage daily attendance of students using both manual and biometric methods.

**1. Writing User Stories**

**a.** *Project Already Chosen: Attendance Management System*

**b.** *5 User Stories (with role, goal, benefit)*

**c.** *Acceptance Criteria’s*

| **User Story** | **Acceptance Criteria’s** |
| --- | --- |
| **1.** As a **teacher**, I want to **mark attendance manually** for each class, so that **records are updated daily**. | Teacher can select class  Can mark each student present/absent  Attendance saves in DB |
| **2.** As a **student**, I want to **see my attendance report**, so that I can **monitor my presence**. | Student can login  Can view total and course-wise attendance |
| **3.** As an **admin**, I want to **schedule holidays**, so that **attendance is not required on those days**. | Admin can select dates  Holidays are reflected in attendance logic |
| **4.** As a **teacher**, I want to **download attendance logs**, so that I can **maintain records offline**. | Can download Excel/PDF  Data is filtered by date or course |
| **5.** As a **student**, I want to **login using fingerprint**, so that **the system auto-marks my attendance**. | Fingerprint scan matches student profile  Attendance auto-updated if verified |

**2. Agile Estimation**

**a. Estimate Story Points using Fibonacci Series (1, 2, 3, 5, 8)**

| **User Story #** | **Task Summary** | **Story Points** | **Reasoning** |
| --- | --- | --- | --- |
| 1 | Manual Attendance Input (Teacher) | 5 | Needs UI + DB integration for class/course mapping and student marking |
| 2 | View Attendance Report (Student) | 3 | Simple UI + fetch data from DB, mostly read operation |
| 3 | Holiday Scheduling (Admin) | 5 | Needs admin UI + calendar logic + DB write operations |
| 4 | Download Attendance Logs (Teacher) | 8 | Export logic to generate PDFs or Excels is complex and time-consuming |
| 5 | Fingerprint Login (Student) | 8 | Biometric integration, hardware access, and auto-attendance logic involved |

**b. Discussion (Reasoning Recap):**

* Stories 4 and 5 have **higher complexity** due to external libraries, hardware integration, or file generation.
* Stories 2 and 3 are moderate in complexity.
* Story 1, though basic, involves repeated interaction per class, hence more effort.

**c. Tasks That May Need Splitting:**

* Story 5 (Fingerprint login) can be split into: Biometric device integration, Profile-linking, Attendance auto-update logic
* Story 4 (Download logs) can be split into: Export to Excel, Export to PDF, Filtered download logic

**3. Agile Metrics and Reporting**

**a. What is a Burndown Chart?**

A **Burndown Chart** is a graphical representation of work left to do vs. time. It shows how quickly the team is burning through user stories. It helps in identifying if the sprint is on track or lagging behind.

**b. Sample 5-day Burndown Chart for Sprint 1 (20 points total):**

| **Day** | **Work Remaining (Story Points)** |
| --- | --- |
| Day 0 | 20 |
| Day 1 | 17 |
| Day 2 | 13 |
| Day 3 | 9 |
| Day 4 | 5 |
| Day 5 | 0 |

*(This shows ideal progress, assuming tasks are completed evenly across the sprint.)*

**c. Other Agile Metrics:**

1. **Velocity:** Measures how many story points the team completes in a sprint. Useful for forecasting future sprint capacity. For example, if Sprint 1 completed 20 points, expect similar in Sprint 2.
2. **Sprint Burnup Chart:** Shows progress toward sprint goals. Unlike burndown, it tracks work completed rather than work remaining.

Helpful when scope changes (i.e., new stories added mid-sprint).