Sweet

STAGE 02 : The report

09.2025

CLAIRE CASTAN

Holberton SCHOOL

Index

task
01

Main objectives
SMART objectives

Project objectives

PROJECT OBJECTIVES

The Sweet application aims to centralize essential information related to diabetes monitoring in order to make patients' daily disease management easier.

It seeks to provide a clear and personalized overview of health status, while enabling patients to prepare for medical appointments with reliable data, and benefit from the support of a peer community.

MAIN OBJECTIVES

1

Centralization and visualization of health data

Provide patients with the ability to record and view their data (blood glucose, weight, physical activity, meals, medication intake + stock, mood).

Offer a clear visual representation (day, week, month) to track progress and better understand trends.

2

Improved medical follow-up

Help patients prepare for their medical appointments with precise, centralized, and historical data.

Provide healthcare professionals with concrete and reliable information (rather than estimates). This facilitates a more accurate diagnosis and adjustment of treatment protocols based on the patient's lifestyle.

3

Organization and personalized monitoring

Provide a calendar to consult or edit historical data and automatically generate weekly and monthly reports.

Allow patients to add notes during medical appointments to improve follow-up and anticipate their needs.

4

Creation of a support community

Provide a forum for patients to share their experiences, difficulties, recipes, or tips.

Combat patient isolation by promoting sharing and solidarity within a community of peers.

SMART OBJECTIVES

1

Centralization and visualization of health data

Enable patients to record and view their daily data (blood glucose, weight, physical activity, meals, medications, and mood) through a functional daily log, operational by week 8 of the project.

2

Improved medical follow-up (daily dashboard)

Provide a clear and ergonomic dashboard allowing patients to view their daily data in real time, delivered and testable by week 8.

3

Organization and personalized monitoring (calendar and reports)

Develop an interactive calendar allowing consultation or modification of historical data and automatic generation of weekly and monthly reports, available and tested before week 9.

4

Creation of a support community (forum)

Deploy a basic forum (topic creation and comments), with at least 80% of functionalities operational no later than week 10, to encourage patient exchanges.

Project Scope

task
02

In-scope
Out-of-scope

PROJECT SCOPE

The Sweet application focuses on simple, reliable, and community-based diabetes monitoring. The project scope defines the features and objectives included in the first version (MVP), as well as those voluntarily excluded to ensure feasibility within the given timeline.

In-scope

MANAGEMENT OF ESSENTIAL HEALTH DATA

Manual entry and consultation of blood glucose data.
Addition and monitoring of weight and physical activity.
Generation of simple statistical reports (average, minimum, maximum).

DASHBOARD AND VISUALIZATION

Synthetic view of data by day/week/month.
Simplified graphs to illustrate trends.

PERSONAL ORGANIZATION

Simplified calendar for medical appointments and data consultation.

Ability to add associated notes.

COMMUNICATION AND COMMUNITY

Basic forum (topic creation, comments).

Articles/news section, managed by the administrator.

MEDICAL CONTACTS MANAGEMENT

Adding essential information about healthcare professionals (name, phone number, note-taking during appointments).

Out-of-scope

Synchronization with connected medical devices (blood glucose sensors, smartwatches, etc.).

Automated medical analysis or advanced medical assistance (automatic diagnosis, medical recommendations).

Advanced social features (private messaging, complex discussion groups).

Multilingual interface (only available in French for the MVP).

Native mobile application (the MVP will be a responsive web app).

09.2025

task
03

End-users
Project team
External stakeholders

Project Stakeholders

END-USERS

DIABETIC PATIENTS

Main actors and sole direct users of the application.

Need:
centralize health data, track progress, and prepare medical appointments.

Full control:
patients decide when and how to show their data to a doctor, with no automatic sharing.

INDIRECT USAGE (VIA PATIENTS)

Healthcare professionals (doctors, nurses, diabetologists)

No direct access to the application.

Patients may decide to present their dashboard or reports during consultations.

Their role is therefore indirect:
using information provided by the patient to adjust treatments and recommendations.

EXCLUSIONS

Relatives and caregivers

No access to the application.

Health data is strictly private, and only patients decide on any potential sharing.

PROJECT TEAM

DEVELOPER & PROJECT MANAGER:
CLAIRE CASTAN

Responsible for backend, frontend, and database development.

Project coordination, timeline management, and deliverable organization.

Supervision of documentation and final presentation.

UI/UX DESIGNER:
CLAIRE CASTAN

Design of the interface and user experience.

Ensuring clear, intuitive navigation adapted to the target audience.

TESTERS / QUALITY:
CLAIRE CASTAN, DIABETIC PATIENT, OTHERS

Verification of developed functionalities.

Bug identification and suggestions for improvements.

Feedback on user experience to refine the application.

EXTERNAL STAKEHOLDERS

EDUCATIONAL SUPERVISORS:
accompany and evaluate each stage of the project.

EVALUATORS/JURY:
assess technical, organizational, and documentary quality during the final presentation.

END-USERS (DIABETIC PATIENTS):
considered stakeholders since the application is designed to meet their needs.

task
04

Technical risks
Organizational risks
User-related risks

Project risks

TECHNICAL RISKS

CHOICE OF TECHNOLOGIES TOO COMPLEX OR UNSUITABLE

Risk: wasted time mastering the stack, project delays.

Mitigation: prioritize known and suitable technologies for the MVP (simple stack, familiar frameworks).

SECURITY AND CONFIDENTIALITY OF HEALTH DATA

Risk: exposure of sensitive data if authentication or storage is poorly managed.

Mitigation: implementation of reliable basic authentication, adherence to best storage practices (even with fictitious data for the project).

LACK OF TECHNICAL TESTING

Risk: bugs during demonstration, loss of credibility.

Mitigation: testing phase planned at the end of the project, testers/ QA involved early in development.

ORGANIZATIONAL RISKS

DELAYS IN SCHEDULING

Risk: some secondary features (forum, articles) not completed on time.

Mitigation: focus on absolute priorities (authentication + health dashboard) before extras.

INCOMPLETE OR SCATTERED DOCUMENTATION

Risk: deliverable not compliant with school expectations.

Mitigation: progressive documentation writing, centralization in a shared folder.

STRESS AND WORKLOAD FOR THE ORAL DEFENSE

Risk: lack of clarity during final presentation.

Mitigation: early preparation of defense (slides + pitch), rehearsals before the deadline.

USER-RELATED RISKS

LIMITED ADOPTION OF THE CONCEPT BY PATIENTS

Risk: perception of the tool as too technical or guilt-inducing.

Mitigation: simple interface, patient-centered, no third-party access (doctors/relatives) to ensure confidentiality.

MISINTERPRETATION OF DATA BY USERS

Risk: patients relying solely on the app without medical advice.

Mitigation: clearly remind that the app is a monitoring tool, not a certified medical device.

09.2025

task
05

Project technologies

PROJECT TECHNOLOGIES

BACKEND – FLASK (PYTHON)

Choice: use of Flask as backend framework.

Justifications:

- Lightweight and modular framework, perfectly suited for an MVP.
- Rich ecosystem (Flask-SQLAlchemy, Flask-Login, Flask-JWT, Flask-Migrate) allowing only necessary additions.
- Easy setup of a REST API to expose data to the frontend.
- Built-in test client, facilitating unit test implementation.
- Performance sufficient for an MVP with limited traffic.
- Prior experience with Flask and Python (leveraging acquired skills).

DATABASE – MYSQL

Choice: relational database MySQL with SQLAlchemy as ORM.

Justifications:

- Stable database, widely used in enterprises.
- Perfectly suited for relational needs (one patient → multiple glucose entries, weights, activities, etc.).
- Standardized SQL language, skills easily transferable to PostgreSQL/MariaDB.
- Native relational constraints (primary and foreign keys) ensuring data consistency.
- Smooth integration with Flask via SQLAlchemy.
- Prior experience with MySQL.

AUTHENTICATION AND SECURITY

Choice: authentication management with JWT (JSON Web Tokens) or Flask-Login.

Justifications:

- Professional best practice, essential when handling personal data.
- Stronger security compared to basic sessions.
- JWT is a market standard, used in many modern applications.
- Scalable: if the app is deployed to the cloud, JWT adapts better than traditional sessions.

Additional measure:

Display of a clear disclaimer in the interface:
“Sweet is a personal monitoring tool. It does not in any way replace the advice of a qualified healthcare professional.”

FRONTEND – REACT

Choice: use of React to build the user interface.

Justifications:

- React is the most in-demand frontend library on the market, strategically aligned with recruiter expectations.
- Reusable components facilitate maintenance (e.g., data entry forms).
- Virtual DOM offers better performance for dynamic dashboards (glucose data day/week/month).
- Mature ecosystem: easy integration of libraries like React Router (navigation) or Chart.js/Recharts (visualization).
- Clear separation between backend and frontend → modern, clean, and scalable architecture.
- Prior experience with React, reinforced and consolidated in this project.

DATA VISUALIZATION – CHART.JS INTEGRATED WITH REACT

Choice: integration of Chart.js to graphically represent glucose, weight, and activity data.

Justifications:

- Lightweight yet powerful library, perfectly suited for an MVP.
- Simplifies patient understanding of data (day/week/month graphs).
- Strong visual impact during the demonstration.
- Customization options (colored thresholds for abnormal values).

PROJECT TECHNOLOGIES

09.2025

CROSS-CUTTING BENEFITS OF THIS STACK

Complementarity:
Flask (backend API) + React (frontend) + MySQL (DB) is a modern architecture, widely used in enterprises.

Realism:
Stack simple enough for an MVP but robust enough for a defense.

Scalability:
Possibility to later add cloud hosting, external APIs, or a mobile app in React Native.

Attractiveness:
strategic choices that highlight skills for a future full-stack developer position oriented towards frontend.

MANAGEMENT AND SUPPORT TOOLS

GitHub:
project versioning, professional standard.

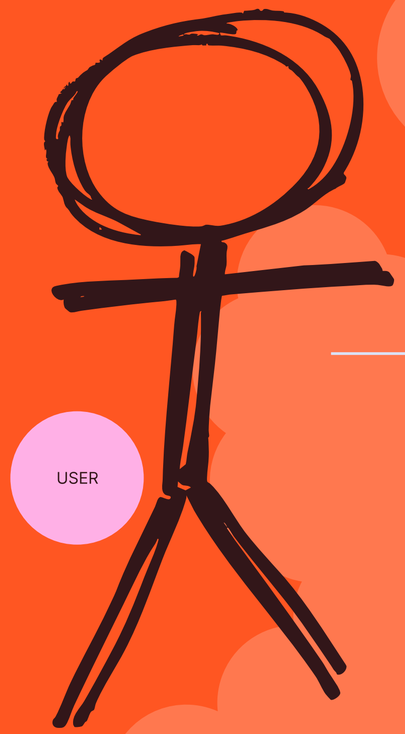
Figma:
UI/UX prototyping before implementation, ensuring a clear and user-adapted interface.

09.2025

task
06

High-level UML diagram

HIGH-LEVEL UML DIAGRAM



Authentication & Account

- Create an account (sign up).
- Log in.
- Log out.

Daily Monitoring (daily form)

- Fill out the daily form with the following items:
  - Blond glucose.
  - Weight.
  - Physical activity.
  - Meals consumed (without calorie calculation, simple description).
  - Medication intake (monitoring + stock management).
  - Mood (on a scale or free comment).
- Modify the daily form several times a day (additions, corrections, adjustments).
- Automatic saving to the database when the day changes.

Consultation & modification of past data

- Open the calendar to select a specific day.
- Consult the form for that day (blood glucose, weight, activity, meals, mood, medications).
- Edit a previous day (e.g., add a missed activity or meal).
- Access weekly and monthly reports directly from the calendar (average, minimum, maximum)..

Organization & medical follow-up

- Add a medical appointment to the calendar.
- Link a healthcare professional to an appointment (example: “Dr Go – diabetologist”).
- Write a personal consultation report (text area for advice, adjustments, or tips given by the doctor).
- Consult the history of medical appointments and their associated notes.

Community & Information

- Create a forum topic.
- Comment on an existing topic.
- Read articles/news published by the administrator.

09.2025

task
07

Conclusion and Final Delivery

HIGH-LEVEL PLAN

STAGE 01

Idea Development → project selection and idea framing.

STAGE 02

Project Charter → definition of objectives, scope, risks, stakeholders, UML. (weeks 3–4)

STAGE 03

Technical Documentation → DB schema, API endpoints, front/back architecture. (weeks 5–6)

STAGE 04

MVP Development (weeks 7–10)

Week 7–8: Development of the daily form + daily dashboard (Objectives 1 and 2).

Week 9: Development of the interactive calendar + reports (Objective 3).

Week 10: Development of the basic forum (Objective 4).

STAGE 05

Project Closure → defense and final documentation. (weeks 11–12)

The Sweet project was designed as a patient-centered pathology monitoring application, with the aim of making daily diabetes management simpler, clearer, and more effective.
The technical choices (Flask, React, MySQL) reflect the desire to combine medical data security, clean architecture, and smooth user experience.

CONCLUSION AND FINAL DELIVERY

The Sweet project was designed as a patient-centered pathology monitoring application, with the aim of making daily diabetes management simpler, clearer, and more effective.

The technical choices (Flask, React, MySQL) reflect the desire to combine medical data security, clean architecture, and smooth user experience.

The system is structured around essential features:

Daily monitoring
(daily form: blood glucose, weight, activity, meals, medications, mood).

Consultation and reports
(via a central calendar).

Medical organization
(appointments, healthcare professionals, consultation notes).

Community
(forum, articles).

Special attention has been given to confidentiality: the patient remains the sole owner of their data, choosing whether or not to share it with their doctor.

The mobile-first design ensures an interface adapted to real patient use and already prepares the transition to a native mobile application. Thanks to this modular architecture (API REST + React frontend), the project can evolve without major redesign into Flutter or React Native, offering a complete mobile experience.

PLANNED DELIVERABLES

Project Charter document (this report).

High-Level UML diagram (Use Cases).

Documentation of technical and functional choices.

FUTURE VISION

The Sweet project aims to go beyond the academic framework to become a truly useful application for diabetic patients in their daily lives.
In the long term, the goal is to provide a solution:

- Secure: respecting confidentiality standards and protecting sensitive data.
- Accessible: mobile-first, intuitive, and adapted to all patient profiles.
- Scalable: able to progressively incorporate new features (integration of connected devices, enhanced medical monitoring, educational tools).

The ultimate goal is for Sweet to be genuinely used by diabetic patients as a centralized monitoring tool, helping them gain clarity and simplicity in managing their condition, and thereby making their daily life easier.