Software Engineering II
Semester 2025-II

Workshop No. 1 — Requirements, User Stories, and Story Mapping

Miguel Angel Buitrago Castillo
Project: VetTrack

Introduction

VetTrack is a veterinary patient management system designed to support small clinics and independent veterinarians in maintaining organized, accurate, and up-to-date patient records. The system facilitates quick access to essential patient information such as medical history, recent visits, treatments, medication schedules, and follow-up needs. By centralizing this information in a clear and structured interface, VetTrack improves workflow efficiency, reduces manual record-keeping errors, and supports faster decision-making during consultations.

One of the common issues in small veterinary practices is that patient information tends to be scattered, some data is on paper forms, some in spreadsheets, some in WhatsApp chats, and some is only remembered by the vet. This makes it harder to track ongoing treatments or recall when a patient was last seen. VetTrack aims to make this process straightforward: the veterinarian logs in, sees the list of active patients, and has all the relevant details in one place. This not only speeds up daily work but also reduces mistakes and helps the clinic maintain a more consistent and professional workflow.

In its initial scope, VetTrack focuses on a single clinic environment and involves two primary user roles: the **Veterinarian**, who interacts directly with patient records and clinical workflows, and the **Admin**, who manages user accounts and basic system settings. While compact in design, the system lays the foundation for future expansion, such as support for multiple clinics, appointment scheduling, and automated reminders, ensuring that VetTrack can evolve alongside the needs of its users.

1. Requirements Documentation

Functional Requirements

A. **Authentication and authorization**
   a. **User login-logout:** The system must allow users to log in with email/username and password, and to log out
   b. **Role-based access:** The system must enforce permissions for **Veterinarians** vs **Admin**

B. **Patient management**
   a. **Create/Read/Update patient:** Create and edit patient profiles: name, species, breed, sex, date of birth/age, color/marks, microchip, owner info (name, phone, email), allergies, notable conditions.
   b. **Search & filter patients:** Search by name, owner, microchip/ID; filter by species, active treatments, last-visit date.
   c. **Patient summary view:** Show at-a-glance: last visit date, reason for last visit, active diagnoses, current treatments/medications, next appointment (if applicable), and recent timeline entries

C. **Clinical records**
   a. **Visits/Consults:** Record a visit, including date/time, reason for visit, vital signs, exam notes, diagnoses, procedures, recommendations, and attachments.
   b. **Treatments & Medications:** Record prescribed treatments/meds: drug dosage, route, frequency, start/stop dates, last administered time, next dose due (optional reminder)
   c. **Active Timeline/Audit:** Automatically log actions on a patient with timestamps and the user.

D. **Media & files**
   a. **Attach media:** Upload and link photos, PDFs, and short videos to visits or patient profiles.

E. **Tasking & follow-ups**
   a. **Reminders/Follow-ups:** Create a simple reminder tied to a patient (e.g. recheck in 2 weeks, next visit date). Show due/overdue in a small list.

F. **Admin & Maintenance**
   a. **User management:** Admin can create/disable users, reset passwords, and assign roles.
   b. **Date Export/Backup:** Export patient/visit date (CSV/JSON) and trigger/download periodic backups.

G. **System & Integrations**
   a. **Basic audit log:** Admin can view a read-only audit (logins, CRUD events)
   b. **Health check endpoint:** Provide a simple /health endpoint indicating uptime and DB connectivity.

Non-Functional Requirements

A. **Performance & responsiveness**
   a. 95th percentile page load <= 2.5 seconds.
   b. Search across <= 10k patients, first result in <= 800ms (server time)
   c. Create/Update patient or visit, server response <= 500ms

B. **Usability**
   a. Key workflows (find patient, open summary, add visit, add medication) are completable in <= 5 clicks
   b. Clear labels and concise forms. Required fields are visibly marked. Inline validation with actionable messages.

C. **Security & privacy**
   a. Enforme RBAC (Veterinarian, Admin).
   b. All traffic over HTTPS
   c. Session timeout after 30 minutes of inactivity; immediate revocation on logout.
   d. Audit trail for create/update/delete on clinical data (who, what, when)

D. **Maintainability & quality**
   a. CI with unit and request-level tests.
   b. Linting and formatting are enforced in CI; API and DB schema versioned with migrations.

E. **Portability & Tech**
   a. Runs in Docker (app+DB)
   b. Zero local OS dependencies beyond Docker and .env configuration.

2. User Stories

Veterinarian (End-User)

a. **Log in/out**
Story: As a veterinarian, I want to log in so that I can securely access my patients.
Acceptance Criteria: Given valid credentials, when I submit the login form, I'm taken to my dashboard. Given I'm logged in, when I click 'Log out', my session ends, and I'm on the login page.

b. **Create a patient**
Story: As a veterinarian, I want to register a new patient so that their data is available for care.
Acceptance Criteria: Given I'm on 'New Patient', when I fill the required fields and save, then the patient appears in search and on my recent lists.

c. **Edit patient info**
Story: As a veterinarian, I want to update patient details so that records remain accurate.
Acceptance Criteria: Given a patient profile, when I edit and save, then the changes are visible on the summary and logged in the timeline.

d. **Search & filter patients**
Story: As a veterinarian, I want to search and filter patients so that I can quickly find a record.
Acceptance Criteria: Given the search bar, when I enter name/microchip/owner or apply filters, then matching results show within one second (server time).

e. **Patient summary at a glance**
Story: As a veterinarian, I want a summary view so that I can see the last visit, active meds, and recent activity quickly.
Acceptance Criteria: Given a patient with data, when I open the profile, I see the last visit, active treatments, next dose (if any), and the activity timeline.

f. **Record a visit**
Story: As a veterinarian, I want to record a consult so that the clinical history stays complete.
Acceptance Criteria: Given a patient profile, when I click "New Visit" and save reason/notes/diagnoses, then the visit appears in history and the timeline is updated.

g. **Prescribe or update medication**
Story: As a veterinarian, I want to add or update medications so that treatment plans are tracked.
Acceptance Criteria: Given a patient, when I add a medication with dosage/frequency/dates, then it appears under active medications with next due (if calculated).

h. **Mark a dose administered**
Story: As a veterinarian, I want to log the last administered time so that adherence is visible.
Acceptance Criteria: Given an active medication, when I mark "Dose given now", then the last administered updates and next due are recalculated.

i. **Create reminders/follow-ups**
Story: As a veterinarian, I want to create a reminder tied to a patient so that I don't miss follow-ups.

Acceptance Criteria: Given a patient, when I set a title and due date, then the reminder appears in my Due/Overdue list on its date.

j. **View Due/Overdue queue**
Story: As a veterinarian, I want a list of due reminders so that I can act on pending tasks.
Acceptance Criteria: Given reminders exist, when I open “Reminders”, then I see due and overdue items ordered by due date.

k. **Activity/audit timeline (read-only)**
Story: As a veterinarian, I want to see who changed what and when so that I can trust the record.
Acceptance Criteria: Given a patient, when I open “Timeline”, then I see chronological entries for create/edit/visit/med changes with the user and timestamp

l. **Attach media**
Story: As a veterinarian, I want to upload photos/PDFs/videos to a visit so that evidence is stored with context.
Acceptance Criteria: Given a visit, when I upload a file, then it’s listed on the visit and downloadable.

Admin.

a. **Manage users**
   Story: As an admin, I want to create/disable users and set roles so that access is controlled.
   Acceptance Criteria: Given the Users page, when I create a user with role Vet/Admin, then the account appears and the role is enforced; When I disable a user, then they cannot log in.
b. **Reset passwords**
   Story: As an admin, I want to reset a user's password so that they can regain access.
   Acceptance Criteria: Given a user, when I click "Reset password" then a reset link or temp password is issued, and the action is logged.
c. **View system audit**
   Story: As an admin, I want to view an audit of logins and CRUD events so that I can monitor compliance.
   Acceptance Criteria: Given the Audit page, when I filter by date/user/action, then matching events (who/what/when) are listed.
d. **Health check visibility**
   Story: As an admin, I want a simple system health indicator so that I can know if the app and DB are up.
   Acceptance Criteria: Given the Status page, when the app is healthy, then I see "OK" with a timestamp; if the DB is unreachable, then I see a clear error.
e. **Enforce password policy**
   Story: As an admin, I want minimum password requirements so that accounts are protected.
   Acceptance Criteria: Given the create/reset password form, when a password doesn't meet policy, then I see validation errors and cannot save.
f. **Data export/backup**
   Story: As an admin, I want to export patient and visit data so that I can create backups or share with stakeholders.
   Acceptance Criteria: Given Export, When I select CSV/JSON and scope, Then a file is generated and downloaded; action is logged.

3. User Story Mapping

Backbone Activity	User Story (Action)	Description	Priority	Release
Authenticate & Access System	Log in / Log out	Veterinarian securely enters the system.	Must Have	MVP
	Role-Based Access	Permissions for Vet vs Admin.	Must Have	MVP
Manage Patient Records	Create Patient	Register new animals.	Must Have	MVP
	Edit Patient Info	Update patient data when needed.	Must Have	MVP
	View Patient Summary	Quick overview (last visit, meds, timeline).	Must Have	MVP
	Search Patients	Search by name/microchip/owner.	Must Have	MVP
Record Medical History	Record a Visit / Consultation	Add medical notes, diagnoses, and procedures.	Must Have	MVP
	Attach Media to Visit	Photos, documents, etc.	Should Have	MVP +1 Iteration
Manage Treatments & Medication	Add/Update Medication	Prescribe treatments.	Must Have	MVP
	Log Medication Dose	Track adherence.	Should Have	MVP +1 Iteration
Follow-Up & Workflow Support	Create Patient Reminder	Schedule re-checks / vaccines.	Should Have	MVP +1 Iteration
	View Reminder Queue	See due/overdue follow-ups.	Should Have	MVP +1 Iteration
Audit / Transparency	Patient Activity Timeline	What changed and when.	Must Have	MVP
Admin Controls	Create/Disable Users	Manage accounts.	Must Have	MVP
	Reset Password	Help users regain access.	Must Have	MVP
	View System Audit Log	System-level audit.	Should Have	Later Release
	Export Data / Backups	Export DB contents.	Should Have	Later Release
System Reliability	Health Check Monitor	Status check for server/DB.	Should Have	Later Release

Prioritization Keys:

a. **Must Have:** Defines de MVP, the system cannot function without it.
b. **Should Have:** Valuable but not required for MVP, add after core workflows.
c. **Could Have:** Optional enhancements, do when time/resources allow.
d. **Won’t Have:** Intentionally out of scope for this phase.