

Sprint 2 review:

Group 4: Kapiainen Casper, Phong Le, Vuorela Tino, Tiihonen Leevi

Done:

Leevi:

Establishing WebSocket connection between back- & front-end. Preparation for Sprint 3 to make drawing real-time between users.

Casper:

Configuring websocket connection at frontend, user creation modal/POST to backend and user creation testing.

Phong :

Design a PostgreSQL database schema for storing drawing data.

Implement a basic API endpoint to query user information by username.

Add backend unit tests with a coverage report.

Tino:

Configuring vitest and creating frontend unit tests, implementing tools for the sidebar.

In progress:

Sprint 2 5 Sep – 16 Sep (9 work items)		0	0	0	Complete sprint	...
<input checked="" type="checkbox"/> QTP-35 Drawing Tools Implementation	FRONTEND CANVAS	DONE				TV
<input checked="" type="checkbox"/> QTP-10 Establish WebSocket connection between frontend and backend	IMPLEMENT CORE F...	DONE				LI
<input checked="" type="checkbox"/> QTP-19 Basic drawing functionality	IMPLEMENT CORE F...	DONE				CS
<input checked="" type="checkbox"/> QTP-43 User creation (send username and profile pic to backend)	FRONTEND	DONE				CS
<input checked="" type="checkbox"/> QTP-14 Create PostgreSQL database schema for storing drawing data	IMPLEMENT CORE F...	DONE				P
<input checked="" type="checkbox"/> QTP-48 basic API endpoint for query data from db - get user by username	BACKEND	DONE				P
<input checked="" type="checkbox"/> QTP-44 Sidebar: user info, color, tool (pen, eraser, hand)	FRONTEND	DONE				TV
<input checked="" type="checkbox"/> QTP-47 Frontend unit tests	FRONTEND	DONE				TV
<input checked="" type="checkbox"/> QTP-46 Backend unit tests and coverage report	BACKEND	DONE				P
+ Create						

Obstacles:

- None so far

Links:

Jira :

<https://metropolia-team-4.atlassian.net/jira/software/projects/OTP/boards/1/backlog>

Github:

https://github.com/PhongLe7de/Software_Engineering_Project_1_Group_4/tree/main

Report recoverage

https://users.metropolia.fi/~phongle/OTP_1_Group_4/Java/

Database structure

Project Folder Structure

```

CREATE TABLE users
(
  display_name VARCHAR(255),
  photo_url TEXT,
  status VARCHAR(20),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE boards
(
  id SERIAL PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  owner_id INT NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT fk_owner FOREIGN KEY (owner_id) REFERENCES users (id)
);

CREATE TABLE user_boards
(
  id SERIAL PRIMARY KEY,
  user_id INT NOT NULL,
  board_id INT NOT NULL,
  role VARCHAR(50) DEFAULT 'EDITOR',
  UNIQUE (user_id, board_id),
  CONSTRAINT fk_user FOREIGN KEY (user_id) REFERENCES users (id),
  CONSTRAINT fk_board FOREIGN KEY (board_id) REFERENCES boards (id)
);

CREATE TABLE strokes
(
  id SERIAL PRIMARY KEY,
  board_id INT NOT NULL,
  user_id INT NOT NULL,
  color VARCHAR(50),
  thickness INT,
  type VARCHAR(50) NOT NULL,
  tool VARCHAR(50),
  x_cord FLOAT,
  y_cord FLOAT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT fk_stroke_board FOREIGN KEY (board_id) REFERENCES boards (id),
  CONSTRAINT fk_stroke_user FOREIGN KEY (user_id) REFERENCES users (id)
);

CREATE TABLE event_logs
(
  id SERIAL PRIMARY KEY,
  board_id INT NOT NULL,
  user_id INT NOT NULL,
  type VARCHAR(50) NOT NULL,
  payload_jsonb JSONB NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT fk_event_board FOREIGN KEY (board_id) REFERENCES boards (id),
  CONSTRAINT fk_event_user FOREIGN KEY (user_id) REFERENCES users (id)
);

```

```

com.otp.whiteboard
├── api
│   ├── controller
│   │   ├── AuthController.java
│   │   ├── UserController.java
│   │   └── DrawController.java
│   └── Endpoint.java
├── config
│   ├── OpenApiConfig.java
│   ├── RedisConfig.java
│   ├── WebSocketConfig.java
│   └── SecurityConfig.java
├── dto
│   ├── auth
│   │   ├── LoginRequest.java
│   │   └── RegisterRequest.java
│   └── user
│       ├── UserByDisplayNameRequest.java
│       ├── UserUpdateRequest.java
│       └── UserDto.java
├── enums
│   ├── DrawEventType.java
│   ├── DrawingTool.java
│   ├── Role.java
│   └── Status.java
├── model
│   ├── Board.java
│   ├── EventLogs.java
│   ├── Stroke.java
│   ├── User.java
│   └── UserBoard.java
├── repository
│   └── UserRepository.java
├── service
│   ├── AuthService.java
│   └── UserService.java
└── WhiteboardApplication.java

resources
├── UserService.java
├── db.migration
│   ├── V1_09012025100350_users_table.sql
│   ├── V1_09032025140330_create_board_and_user_board_table.sql
│   └── V1_09032025140330_create_stroke_and_event_logs_table.sql
├── application.properties
├── application-local.properties
└── templates / static

```