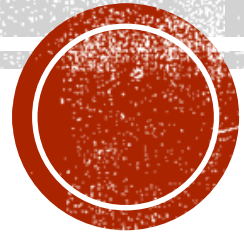
BOOKLY

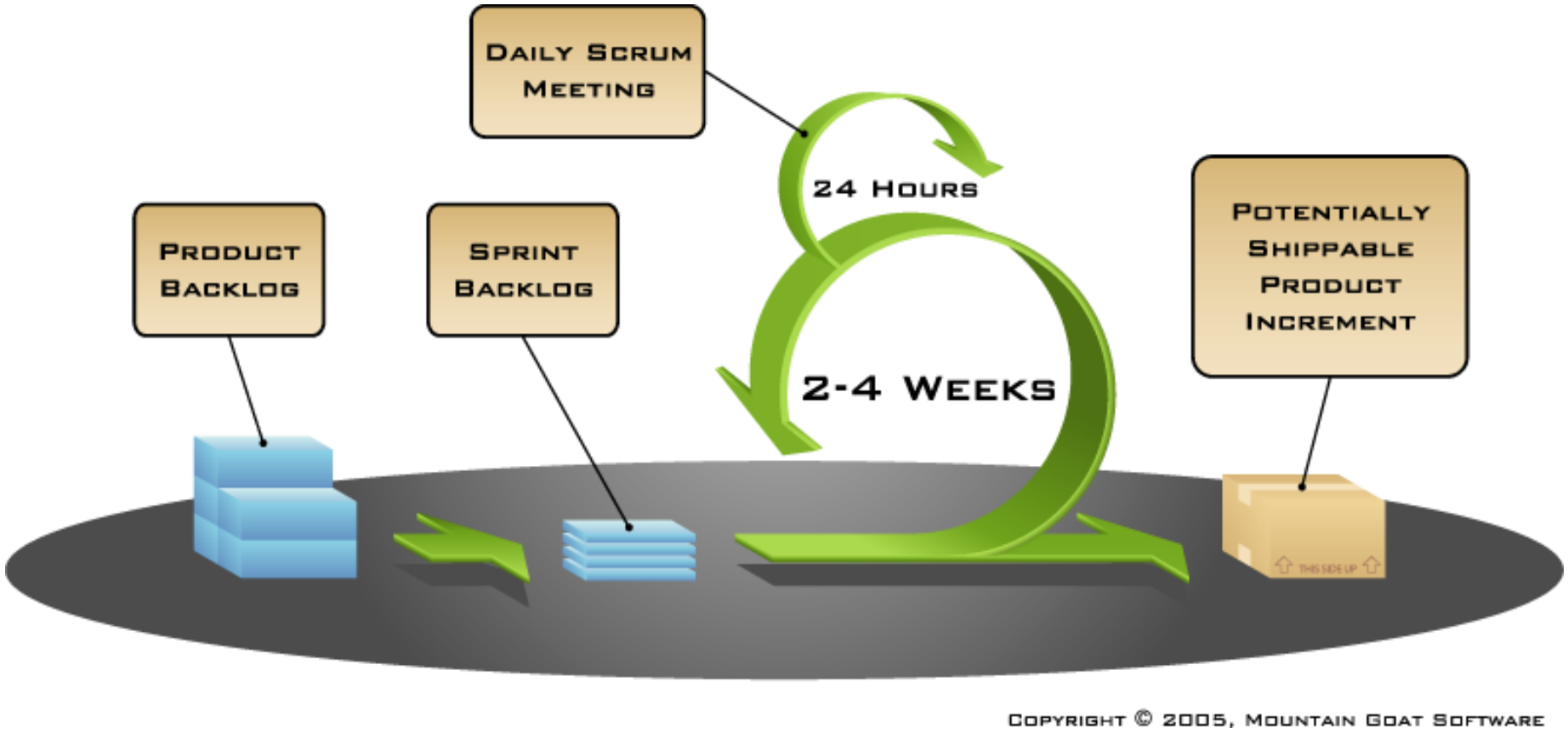
midterm presentation



OUTLINE

- Project Management
  - Scrum and YouTrack
  - Burn-Down-Chart
  - RUP
  - Extend of functionality
  - Use Cases for Mockup
- Architecture
- Automation
- Demo

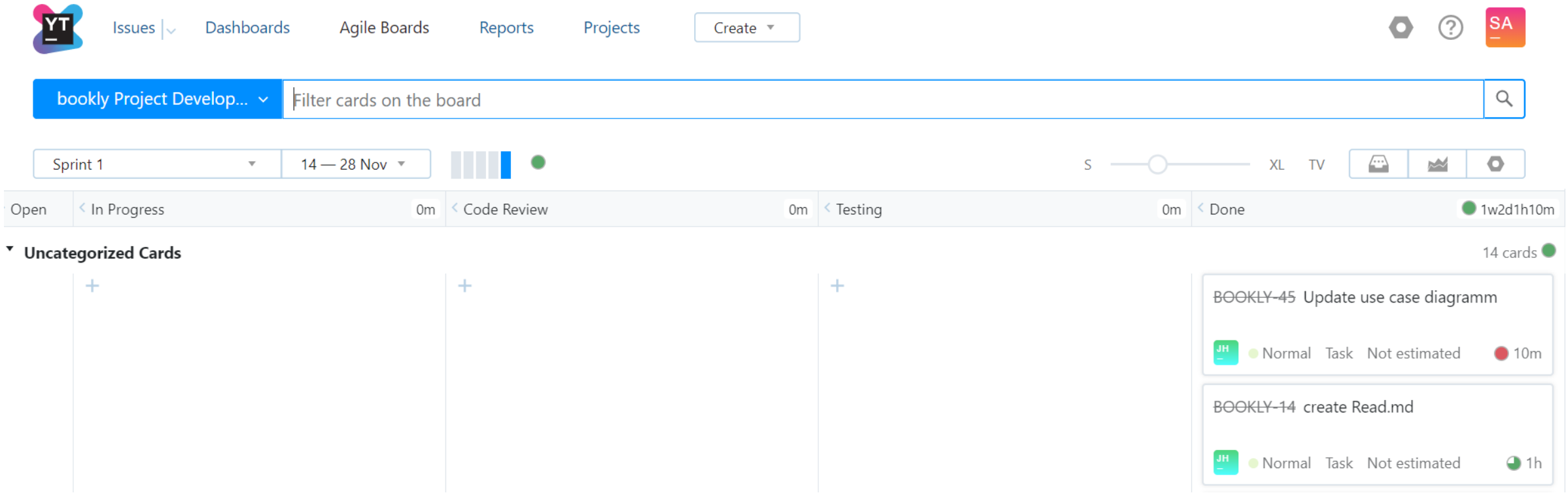
SCRUM AND YOUTRACK



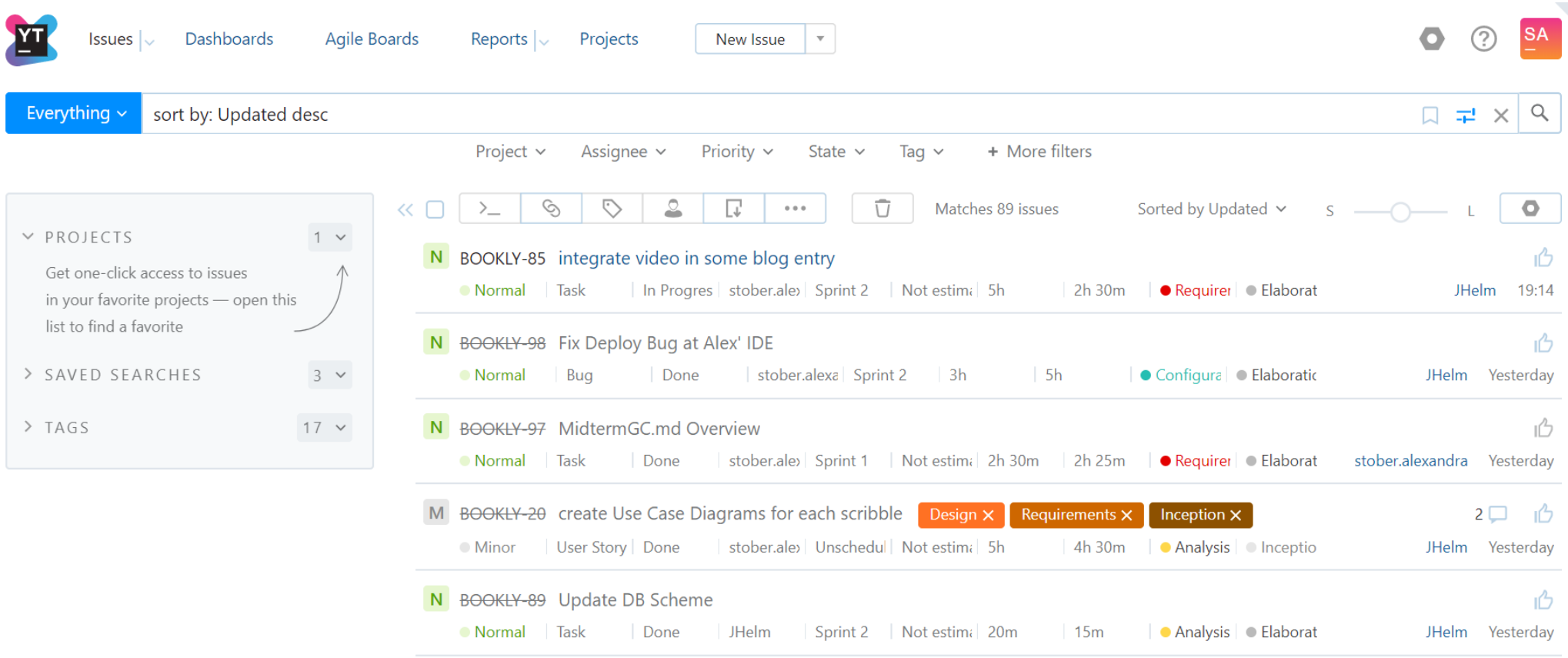
RUP- RATIONAL UNIFIED PROCESS

- **Alexandra**
  - **Requirements Specifier:** Describe and phrase the features / design of the website with the team
  - **Project Manager:** Phrase acceptance criteria of the features + Coordinate sprints + Manage YouTrack + Assign Tags
- **Jeanne**
  - **Implementer:** Create database structure, logic, and interfaces (API)
  - **Tester:** Test the interfaces (APIs) between Frontend, Backend and Database and the functionality of the Frontend
- **Nico**
  - **Deployer:** Host domain and integrate CI (and Docker)
  - **Designer:** Implement the design described in the Use-Cases + Web tests

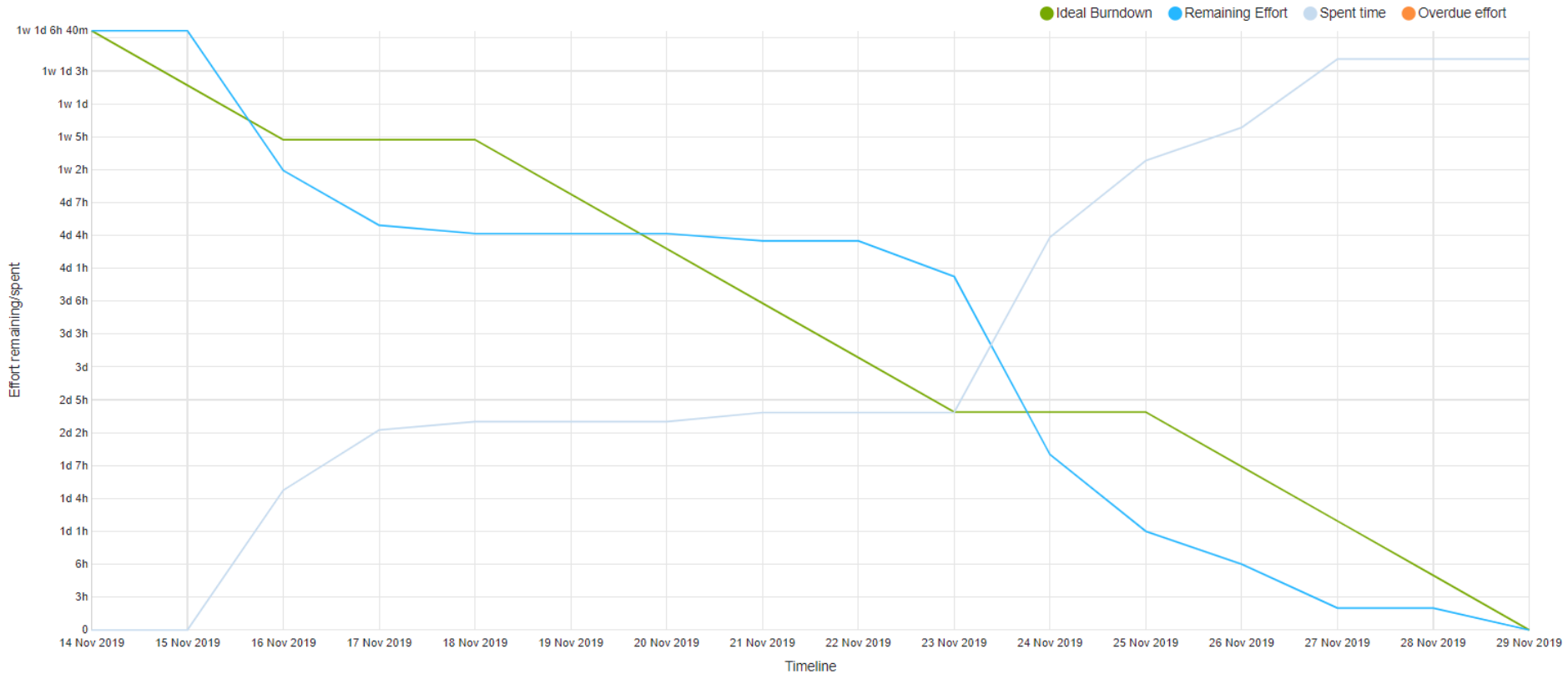
SCRUM AND YOUTRACK



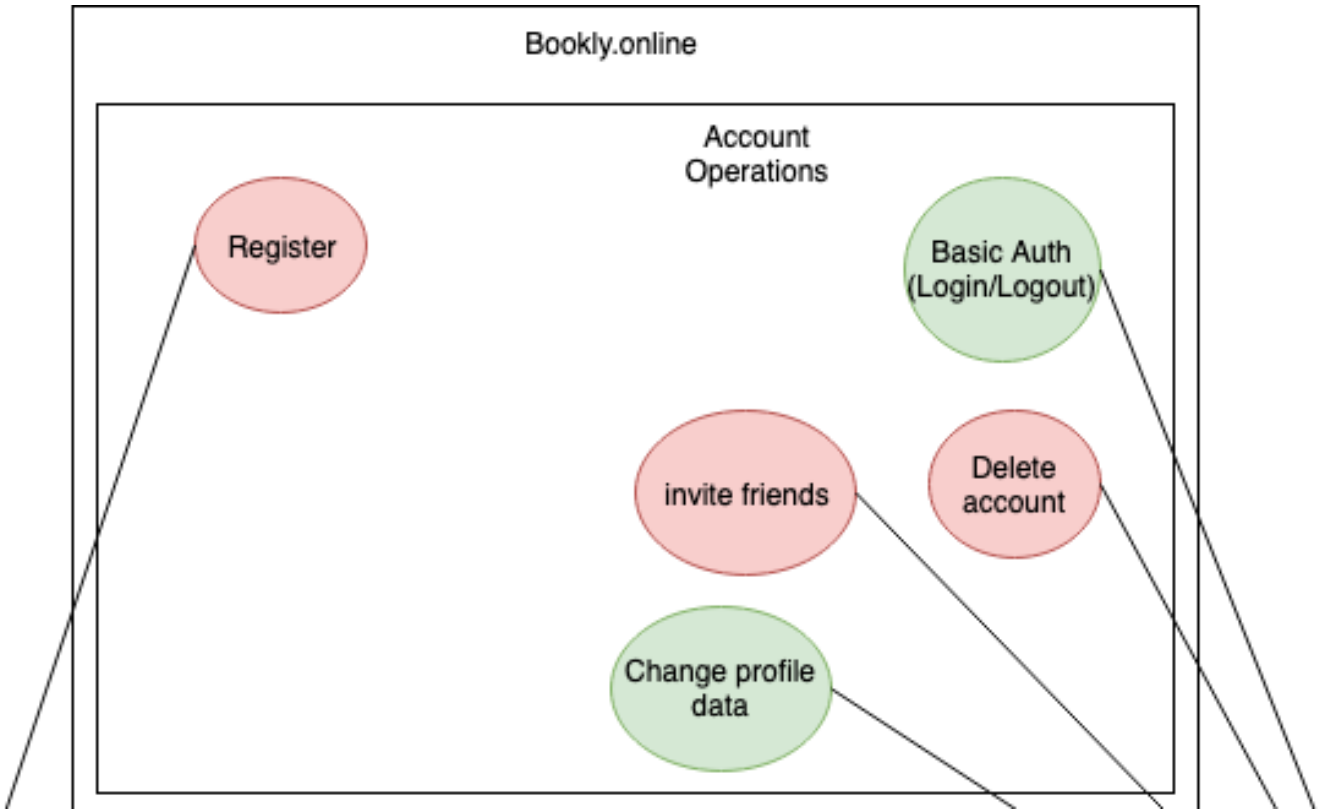
SCRUM AND YOUTRACK

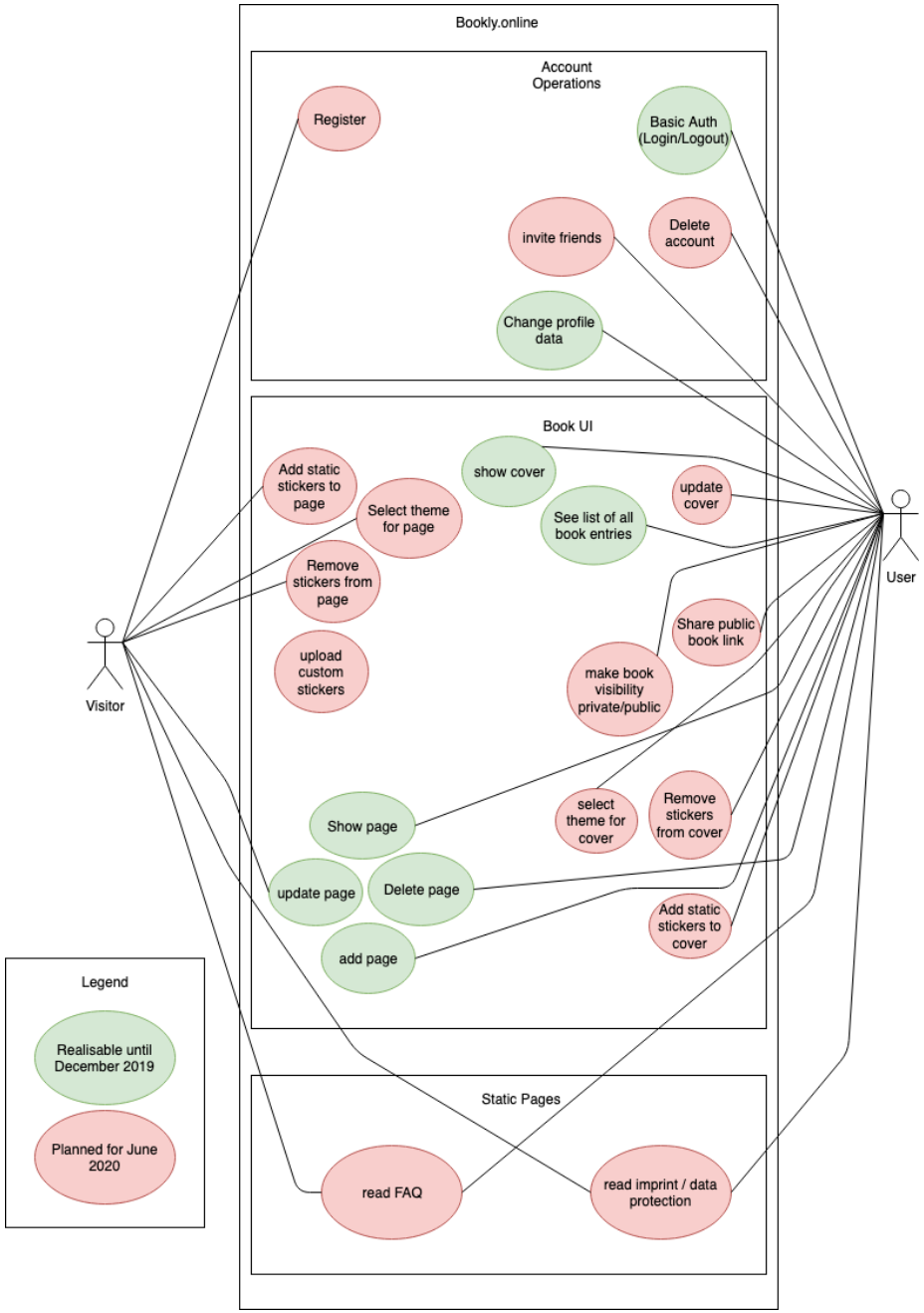


BURN-DOWN-CHART

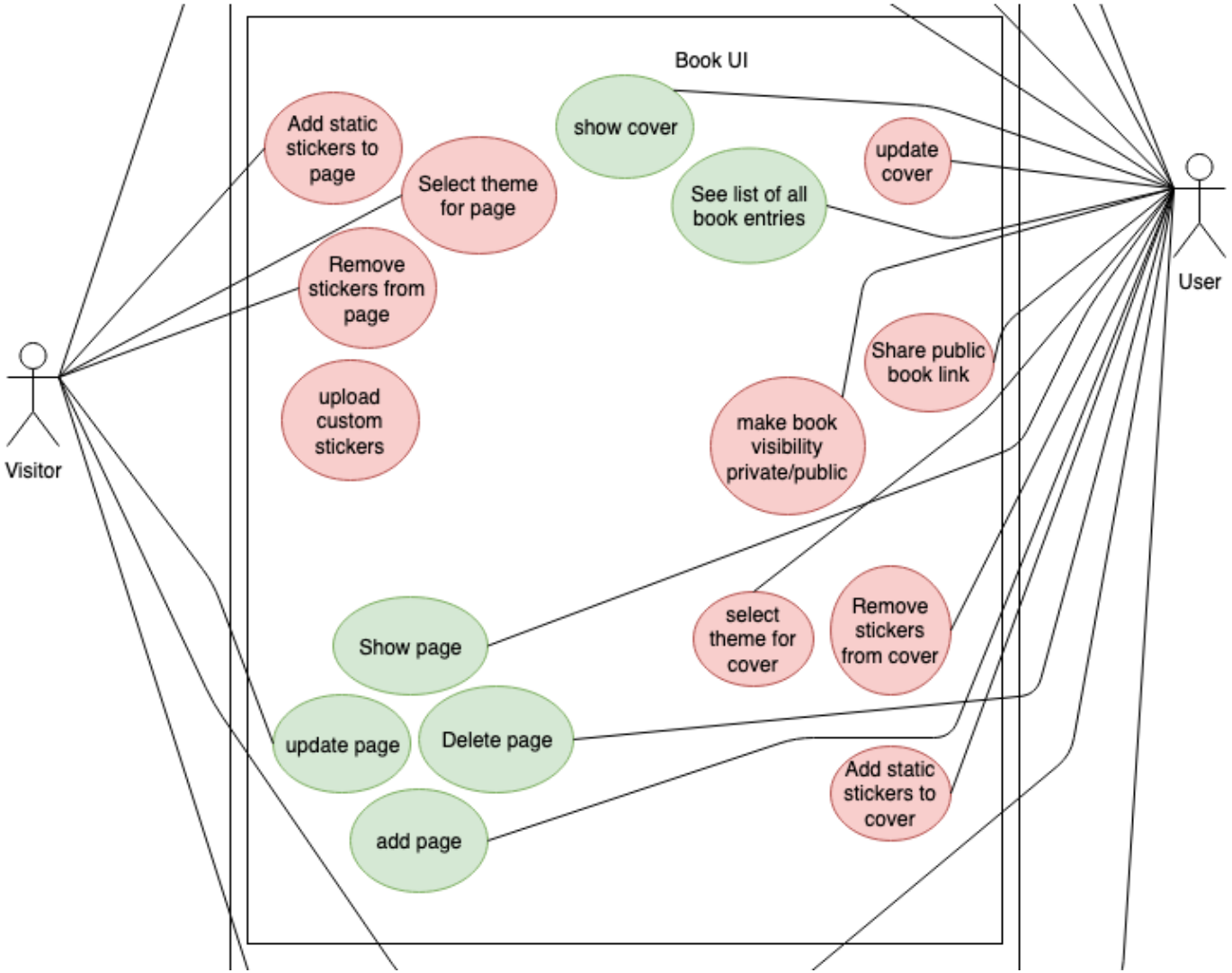


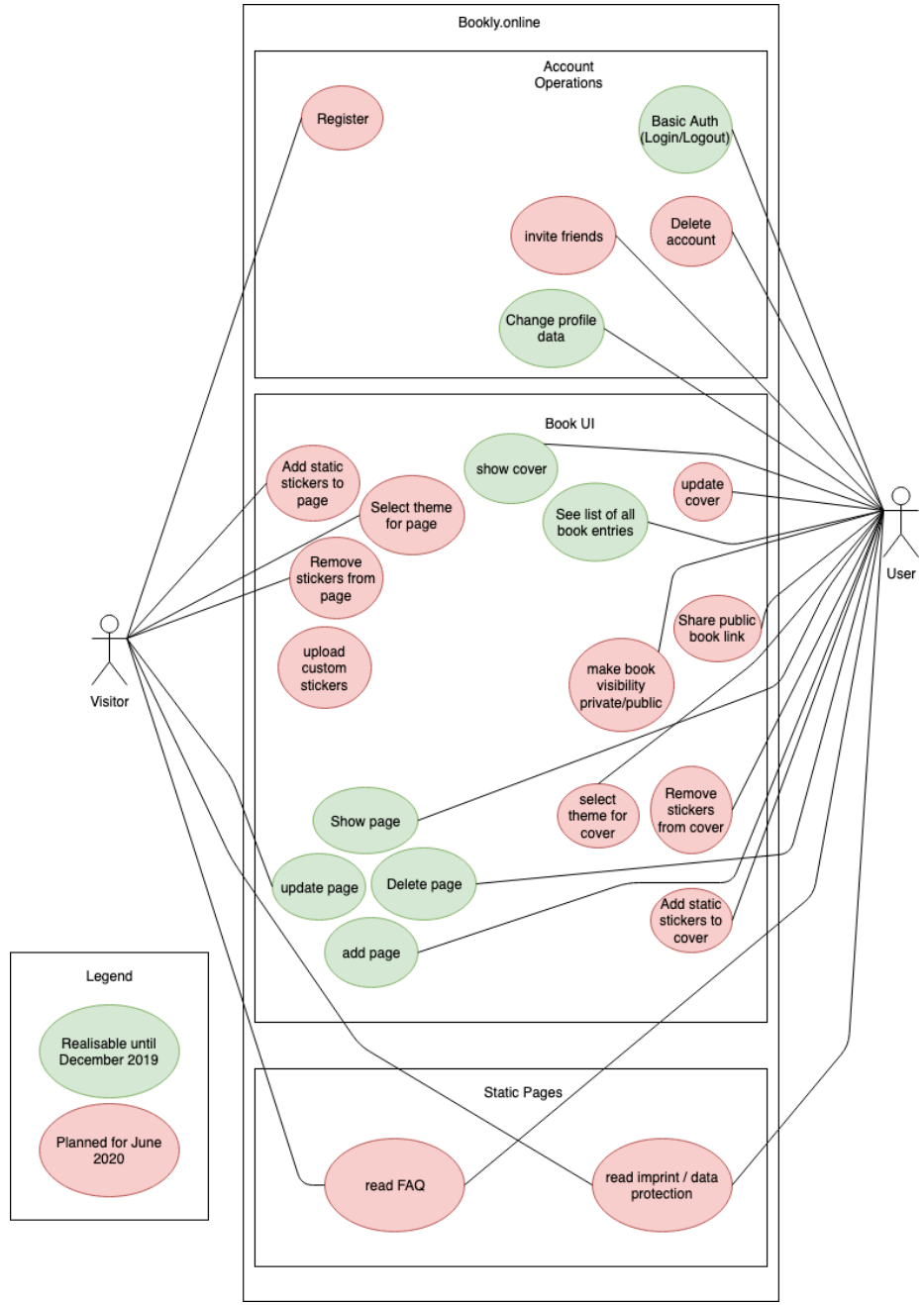
EXTEND OF FUNCTIONALITY





EXTEND OF FUNCTIONALITY



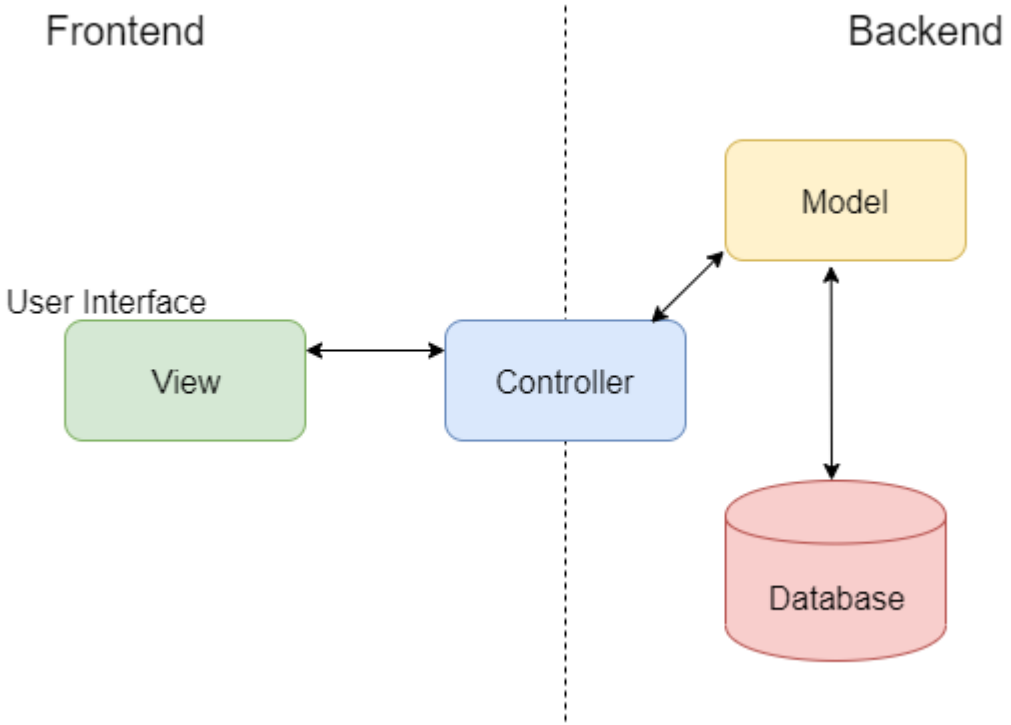


USE CASES FOR MOCKUP

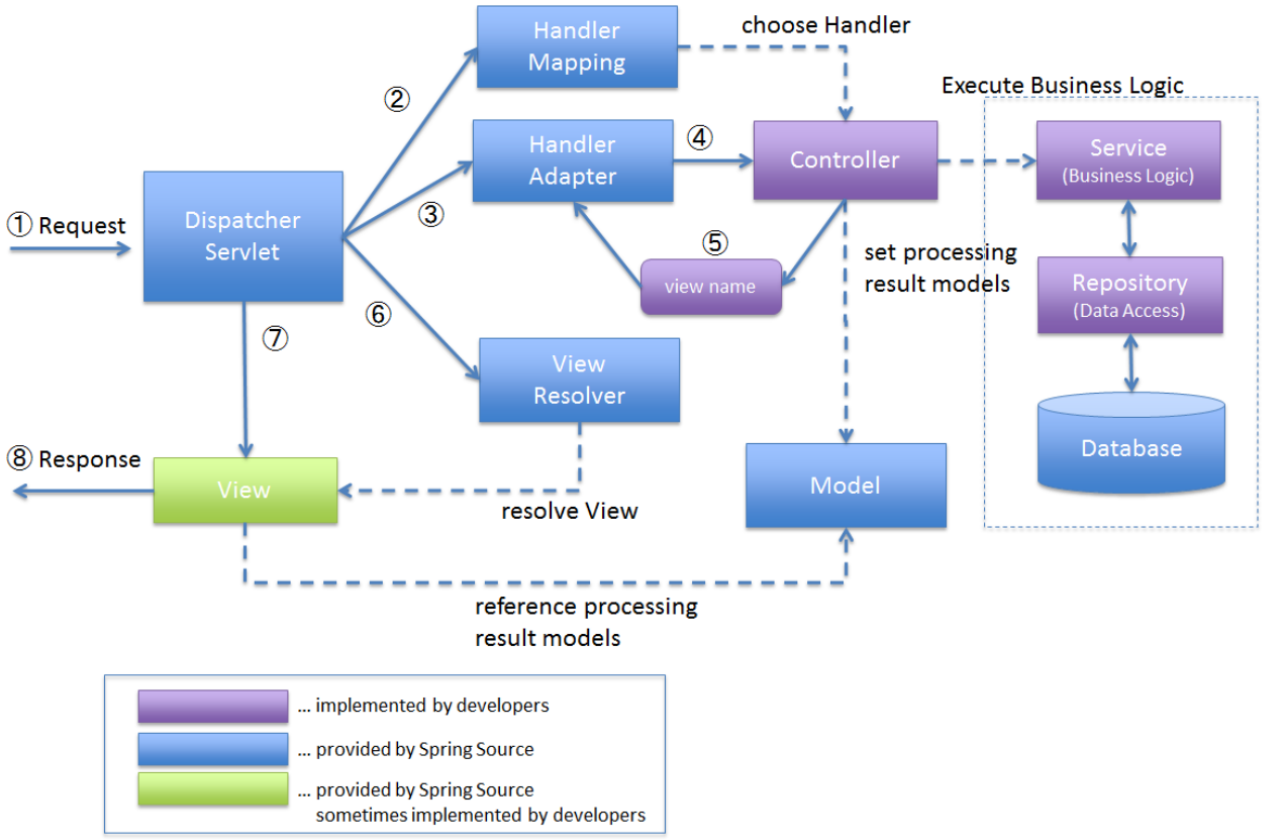
- CRUD: Manage page:
  - C: create book entry - S E
  - R: read an entry - S E
  - U: update/edit an entry - S E
  - D: delete an entry - S I
- CRUD: Manage Book:
  - C: create a book - S E
  - R: show all entries & cover - S E
  - U: update the cover - X I
  - D: delete the whole book - S I

Legend:
S=Scope; X = out of scope;
I = Included in CRUD (not counted as UC)
E = Excluded (external UC document -> counted as UC)

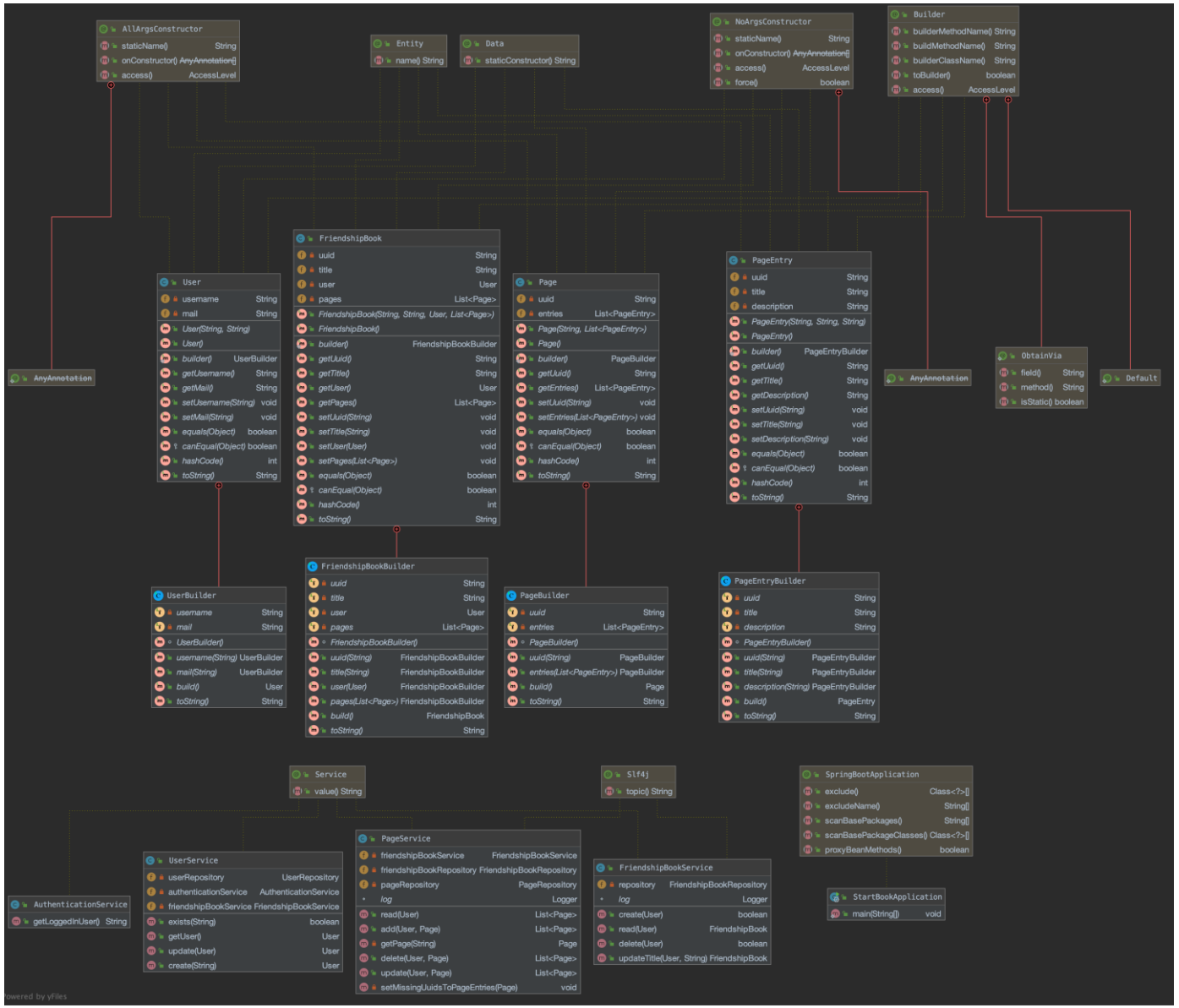
ARCHITECTURE - OVERVIEW



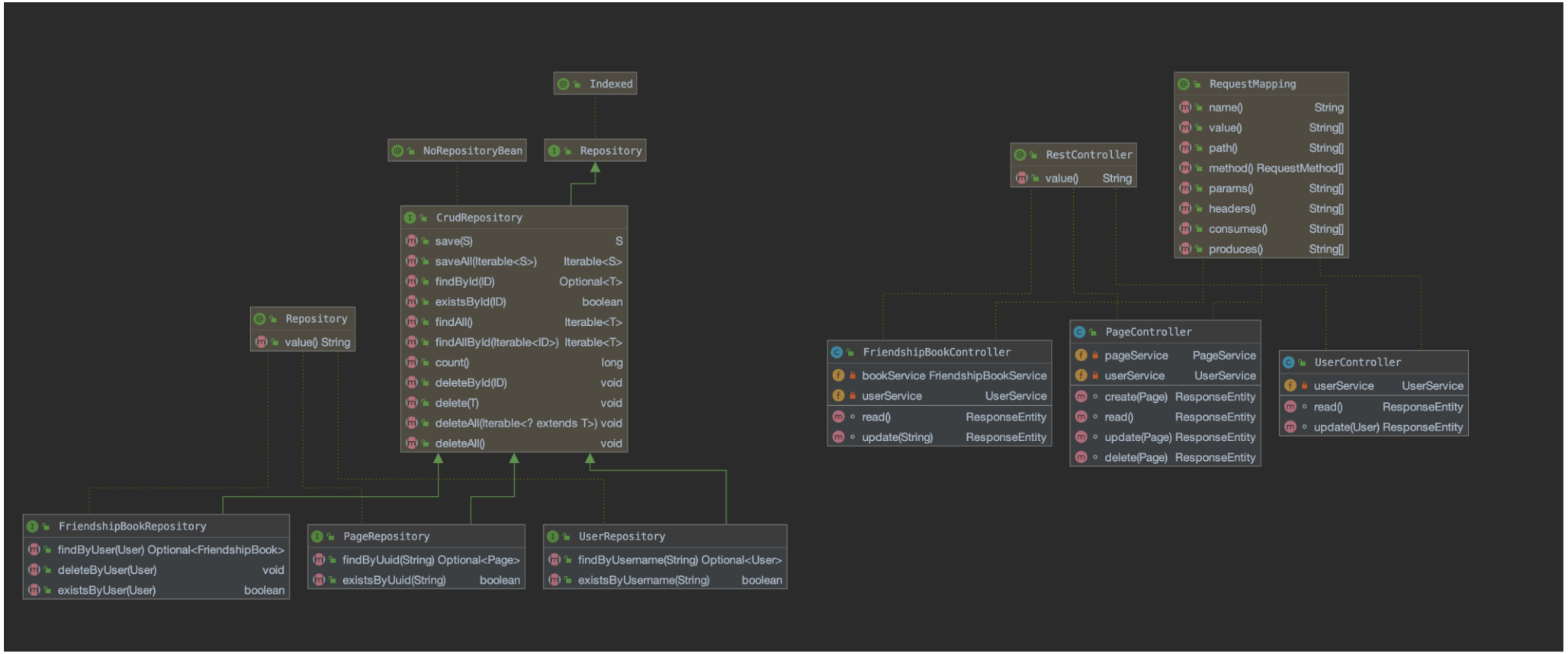
ARCHITECTURE – OVERVIEW



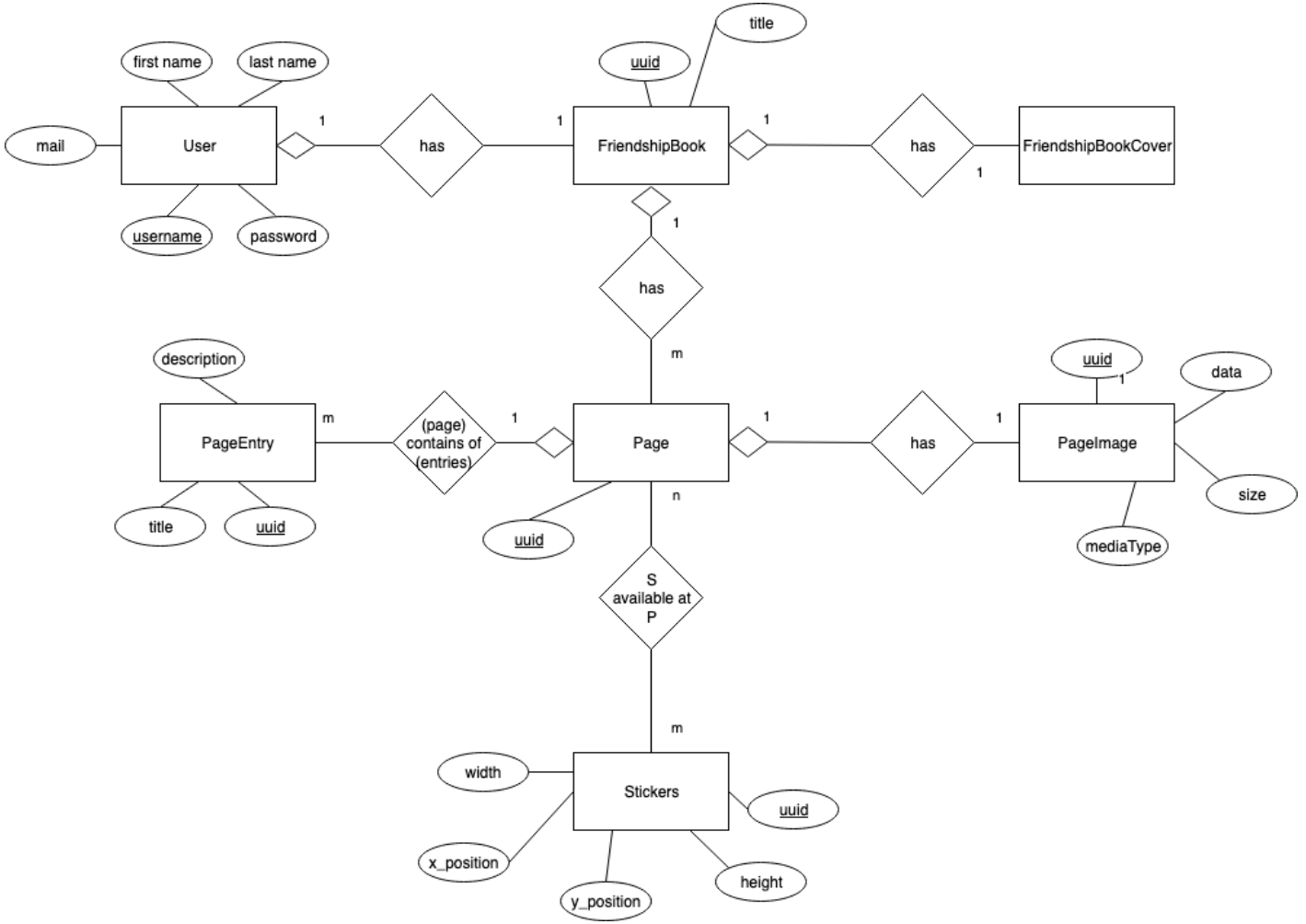
CLASS DIAGRAM



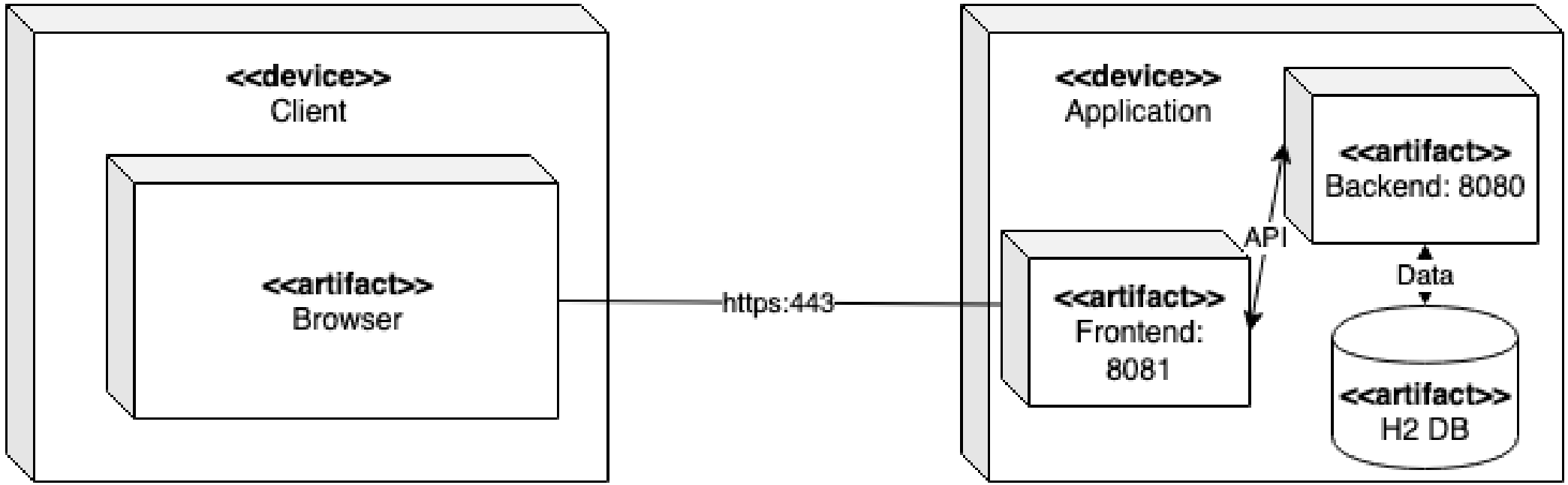
CLASS DIAGRAM



ARCHITECTURE – DATA VIEW



ARCHITECTURE – DEPLOYMENT VIEW



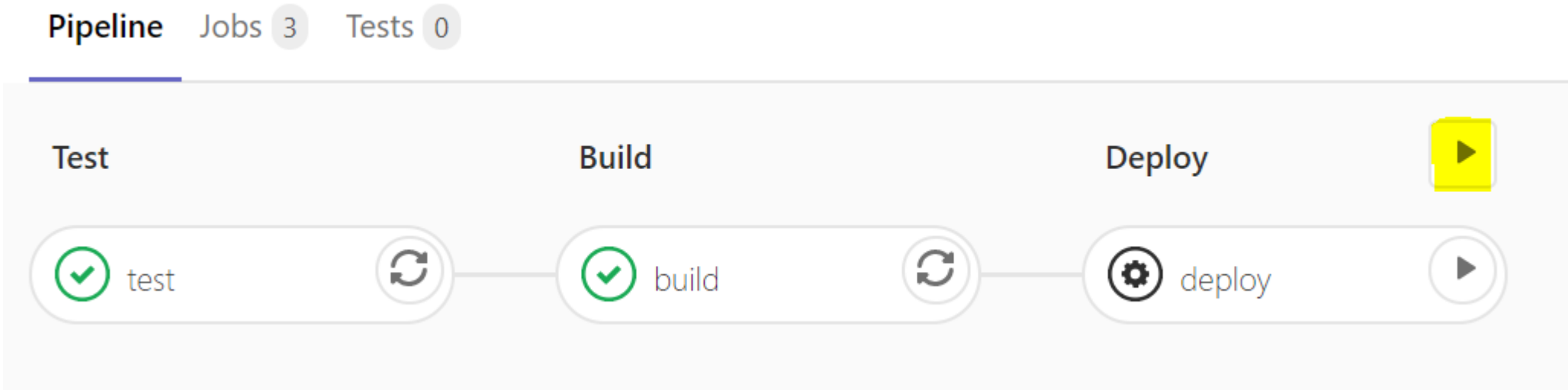
AUTOMATION

- BDD tests with Cucumber are up and running

```
public class Stepdefs {

    private String answer;

    @Given("I login with {string}{string}")
    public void i_am_logged_in(String username, String password) throws Exception {
        TestRestTemplate testRestTemplate = new TestRestTemplate();

        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.MULTIPART_FORM_DATA);
        String encodeBytes = Base64.getEncoder().encodeToString((username + ":" + password).getBytes());
        headers.add(HttpHeaders.AUTHORIZATION, headerValue: "Basic " + encodeBytes);

        MultiValueMap<String, Object> body = new LinkedMultiValueMap<>();

        HttpEntity<MultiValueMap<String, Object>> request = new HttpEntity<>(body, headers);
        ResponseEntity<String> response = testRestTemplate.exchange(
                url: "http://localhost:8080/api/user", HttpMethod.GET, request, String.class,
                new LinkedMultiValueMap<>());
        answer = response.getBody();
        System.out.println(answer);
    }
```

AUTOMATION

- Pipeline: automatic Test -> Build -> Deploy with one click



DEMO

BOOKLY - friendship book