SATELLID: PERSONAL KNOWLEDGE MANAGER

with Web Technologies

Muhammad Haidar Hanif

18 March 2015

Gunadarma University

OUTLINE

Introduction : Background, Problem Statements, Objectives, Scope

Literature Study : KMS, SDLC, Design Pattern, Database, Programming, Platform, Framework, Soft. Testing, SCM

Analysis & Design : Target User, User Types, User Stories, Contextual System, Functionality Flow, App Architecture, User Interface & Interaction

Implementation & Testing : Agile Development, Code, and Test

Conclusion & Suggestion

INTRODUCTION

There are tons of daily personal knowledge that must be managed,
but can’t be handled with regular tools.

How to manage tons of daily personal knowledge we have with just a simple and single system of knowledge manager that implemented with Web technologies?

Background

Solve a problem around *daily knowledge management*.
Need for tool and system to make it more effective. ...

Problem Definition

1. How to manage tons of personal or even collective knowledge we have with just a simple and single system of knowledge manager?
2. How can knowledge manager naturally structure the data into knowledge that has context?

Objectives

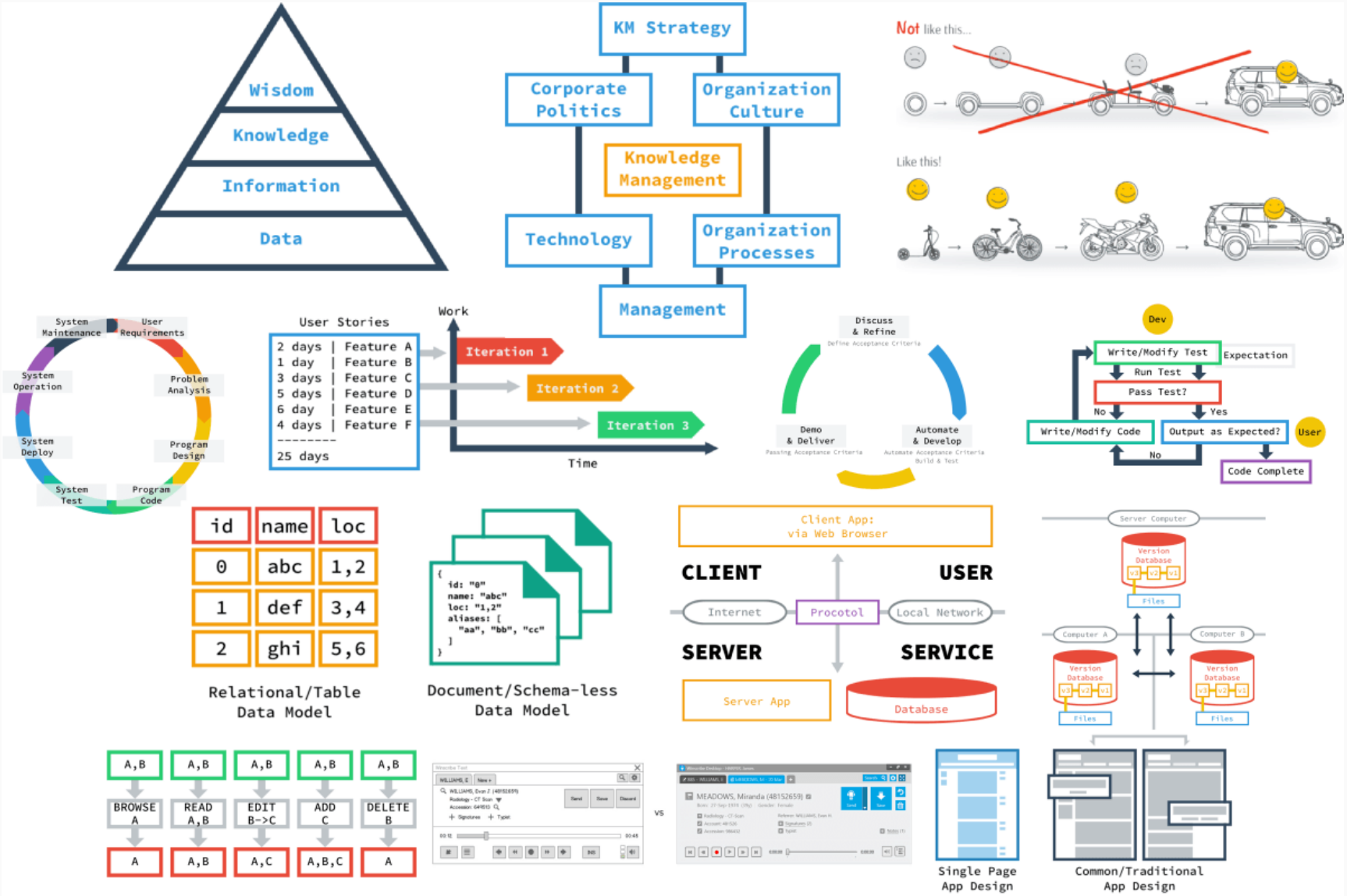
- Define and develop a new kind of knowledge management system (KMS) or knowledge manager called “Satellid knowledge manager” to met the solution.
- It utilize knowledge template which use context and structure, to store and managing personal daily knowledge.
- Developed and built using Web technologies.

Scope

- Create a *simpler system* to do daily knowledge management for personal use with basic BROWSE, READ, EDIT, ADD, DELETE (BREAD) features.
- The main methodologies are agile, MVP, and ATDD.
- The tools are MongoDB, JavaScript/CoffeeScript, JSON, Node.js, Meteor framework, and Git.

LITERATURE STUDY

LITERATURE STUDY



mongoDB

CouchDB
relax

JS

{JSON}

node

io

METEOR

VELOCITY

git

GitHub

Knowledge Management System

Data-Information-Knoweldge-Wisdom (DIKW),
Knowledge Management System (KMS), Personal Knowledge Manager

Software Development Life Cycle

Agile methodologies, Minimum Viable Product (MVP),
Acceptance Test Driven Development (ATDD)

Design Pattern

BROWSE, READ, EDIT, ADD, DELETE (BREAD),
simple interaction and interface design with mockup,
and Single Page Application (SPA)

Database

NoSQL, document database, MongoDB

Programming

JavaScript, JSON, Node.js, web application, framework, full stack framework called Meteor, source code management (SCM) with Git

ANALYSIS & DESIGN

SATELLID

Target user:

Person who frequently gather and need to manage their knowledge at almost everytime.

User Types:

New user, existing user, regular person, researcher, engineer, developer, designer, information architect, event speaker, student, teacher, leader, collaborator, colleagues, recruiter, etc

Agile User Stories

1. As a System, I need to be run on supported platform and via a network
2. As a System, I can have the data imported without the app opened
3. As a User, I want to use the app via web browser
4. As a User, I want to read knowledge that already stored
5. As a User, I want to search a knowledge and browse the search result
6. As a User, I want to add a new knowledge based on context
7. As a User, I want to delete a stored knowledge
8. As a User, I want to edit a stored knowledge

Contextual System

A data and template approach to classify the knowledge with its context and structure.

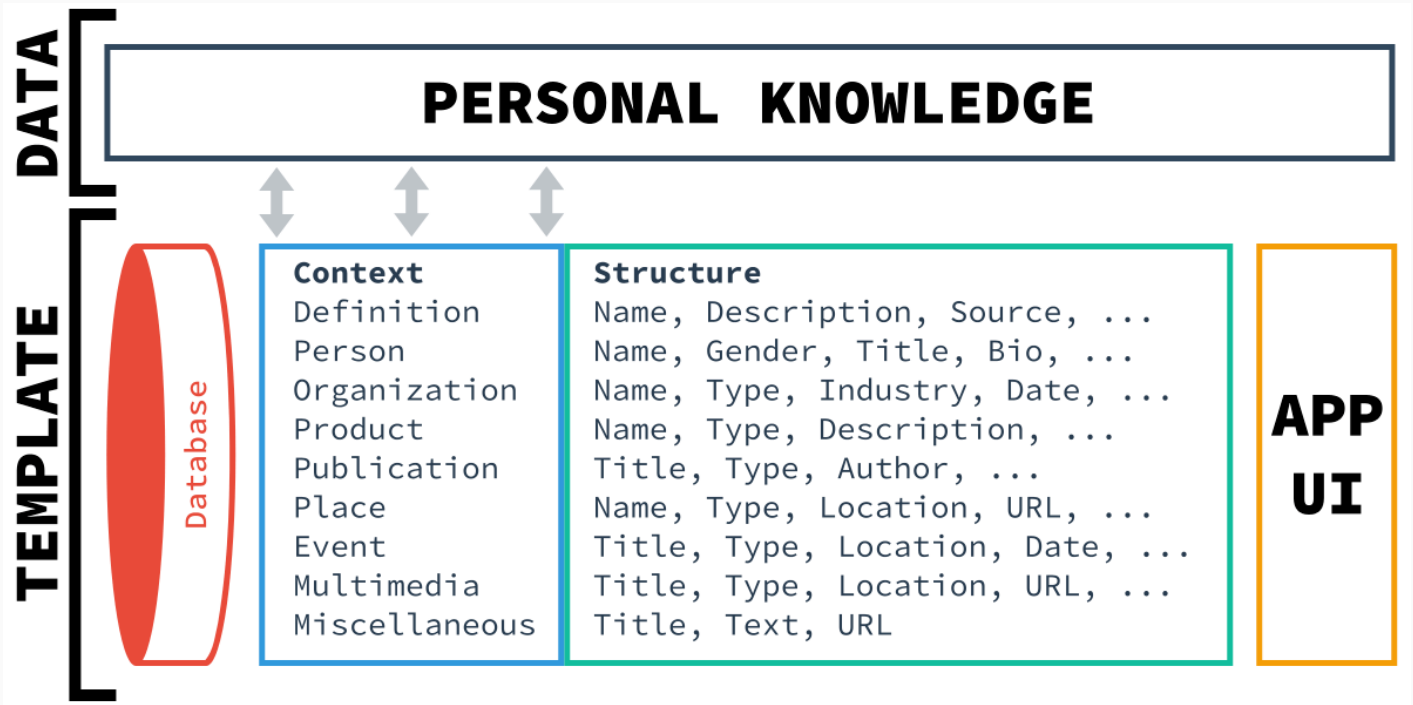


Figure 1: Contextual System

Functionality Flow

1. Turn on and turn off the System via terminal/CLI.
2. Access and use the System via Web browser.
3. Type and search in the search bar that could automatically show the search result.
4. Browse the existing knowledge or search result in the collection of cards.
5. Add a knowledge that will show the form box to fill.
6. Edit a knowledge with edit mode of form box.
7. Delete a knowledge.

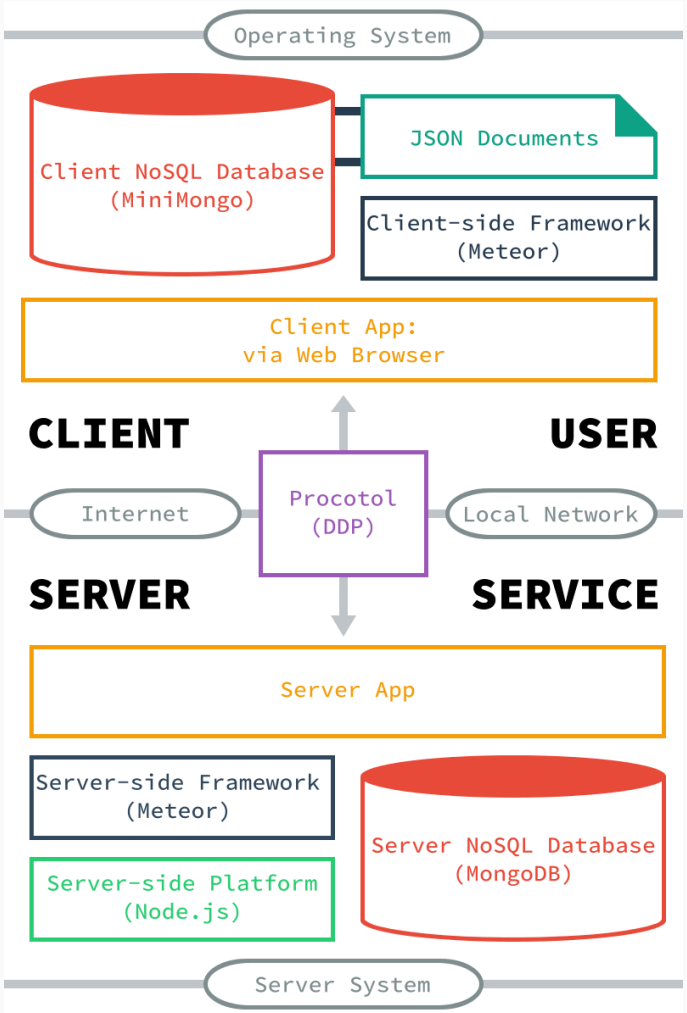


Figure 2: Application Architecture

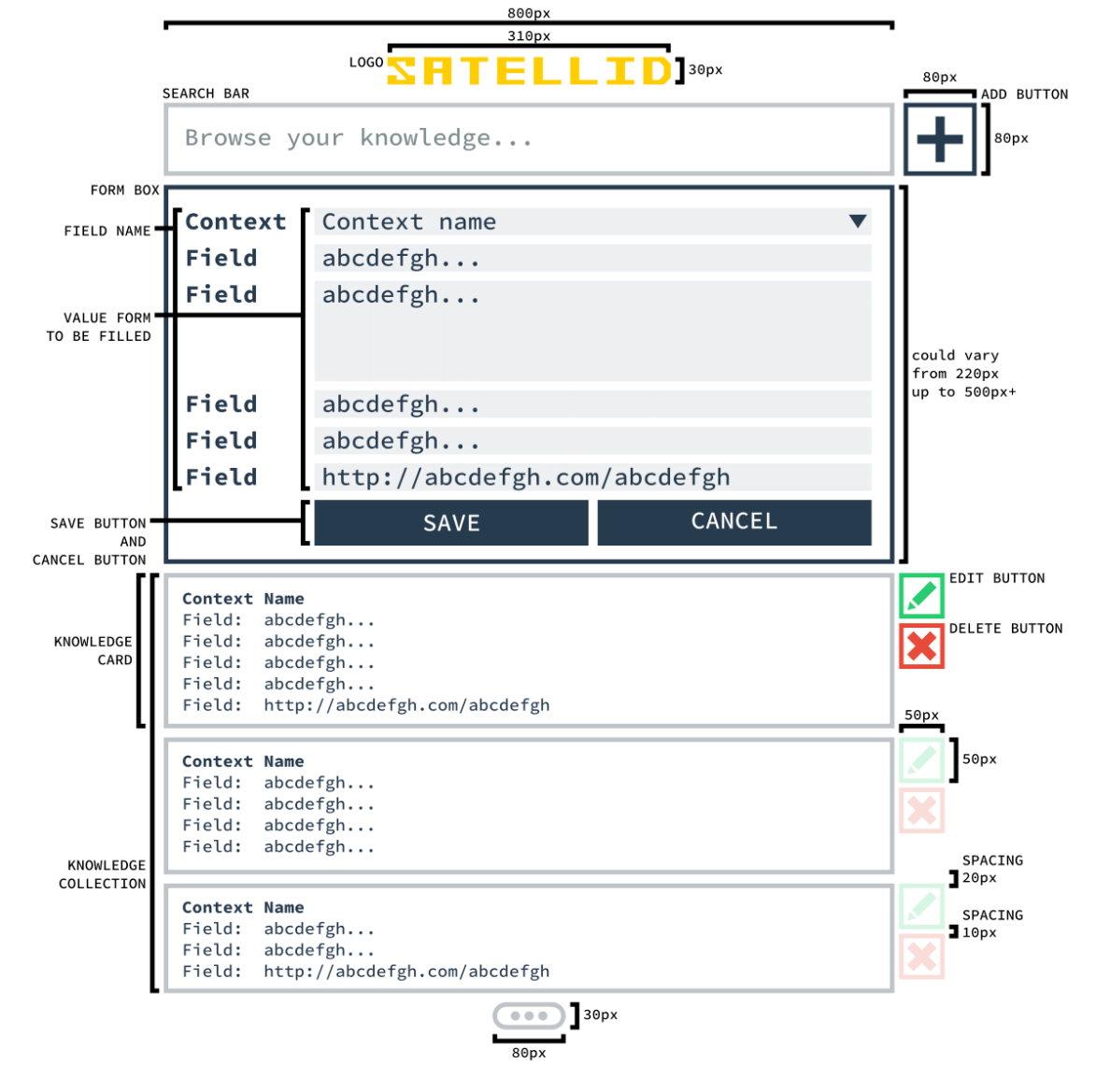


Figure 3: User Interface Mockup Design

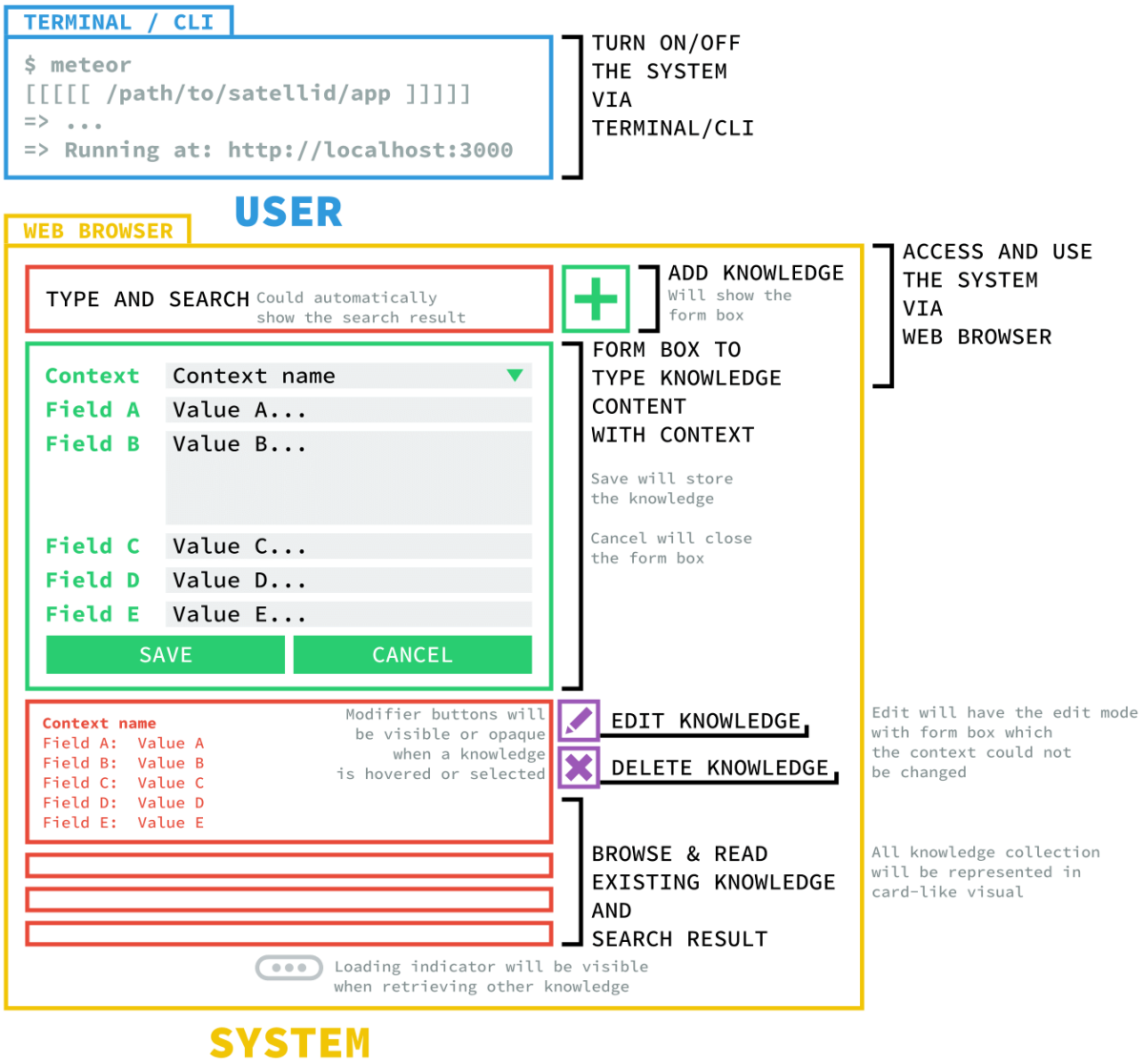


Figure 4: User Interaction Design

IMPLEMENTATION & TESTING

Snippets of Server Code

```
if (Meteor.isServer) {
  Meteor.startup(function() {
    ...
  });
}
```

Snippets of Client Code

```
if (Meteor.isClient) {
  Template.add.events({
    "click button": function() {
      ...
    }
  });
}
```

Figure 5: Application Result Screenshot

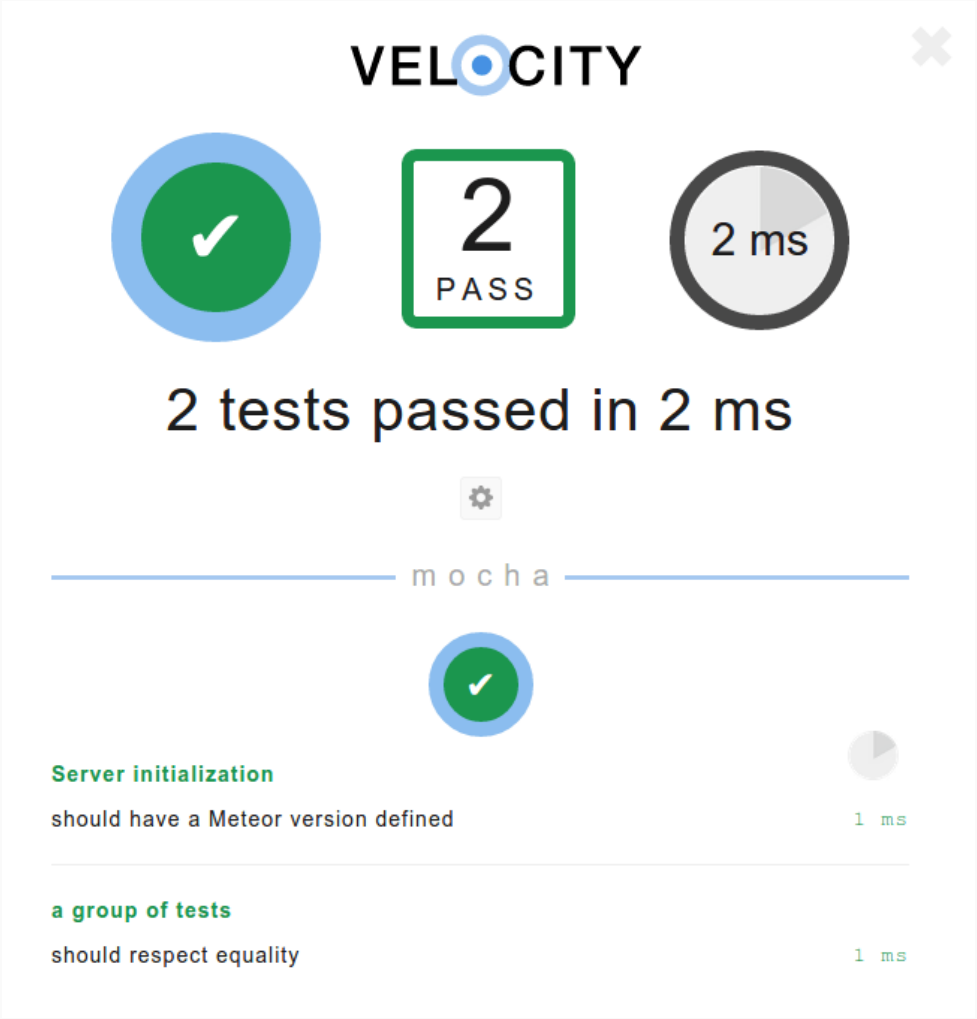


Figure 6: Testing Screenshot

CLOSING

CONCLUSION

1. Simple and single system of managing tons of personal daily knowledge
2. Implementation with Web technologies
3. Flexible data schema combined with template
4. Template based on context

SUGGESTION & FUTURE WORK

There could be some inadequacy, so further and iterated improvement will remain to be done continuously

Future Feature Roadmap:

Account system, easy import & export, more predefined context and field, BREAD the template, more custom configuration, multimedia support, integration with other networks, encryption, etc

Thank You