SATELLID: PERSONAL KNOWLEDGE MANAGER

with Web Technologies

Muhammad Haidar Hanif

18 March 2015

Informatics Engineering - Gunadarma University

OUTLINE

Introduction

Literature Study

Analysis & Design

Implementation & Testing

Conclusion & Suggestion

INTRODUCTION

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

Daily personal knowledge are things like details or facts of:

People Profiles and contacts

Organizations, Schools, and Companies

Glossary of Definition

Hardware and Software Product Listing

Publication of Papers, Books, News, or Presentations

Known or Visited Places or Locations

Upcoming and Attended Events or Schedule

Multimedia Info such as Photo, Music, or video

Lot of Other Miscellaneous Things

All of these are also the context of knowledge

There are a lot of tools/systems for each, that contains knowledge

Contacts in Gmail, Skype
Apple Contacts

Facebook, Google+

LinkedIn, CrunchBase

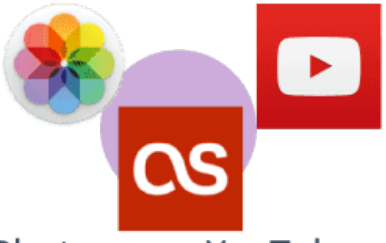
Wikipedia, Silk,
Wordnik

Product Hunt, Engadget
StackShare, AlternativeTo

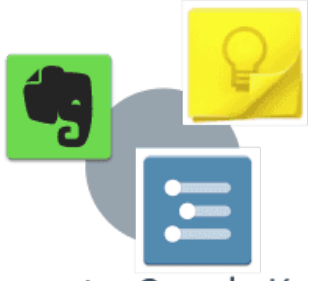
Academia.edu,
Google Drive and Docs

Google Maps, Foursquare, Tripit

Lanyrd, EventBrite,
Google Calendar



Photos app, YouTube,
Last.fm



Evernote, Google Keep,
Workflowy

There is a need for a single personalized system

How to manage those daily personal knowledge with single system?

PRIMARY BACKGROUND

- Solve a problem around daily knowledge management with a system that is personal.
- Essential knowledge are mostly separated into too many softwares.
- Need for a simple and single personalized system.
- Managing knowledge should use a knowledge manager.

PRIMARY OBJECTIVE

Define and develop a personal knowledge manager named “Satellid”, that built using Web technologies.

PROBLEM DEFINITION

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

PROBLEM SCOPE

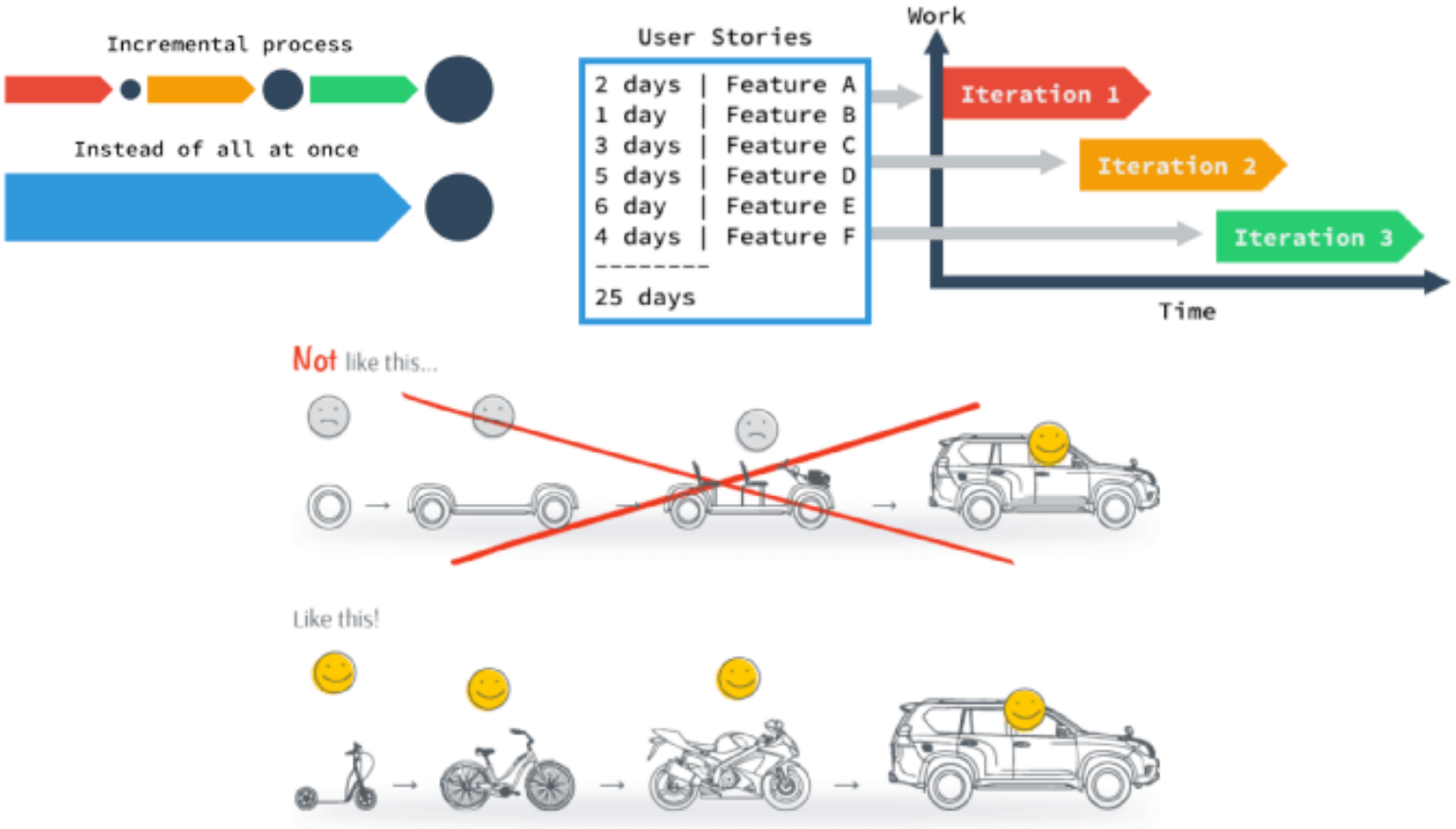
- Create a simple system to do daily knowledge management for personal use
- The main methodologies are agile, MVP, and ATDD.
- The main technologies are MongoDB, JavaScript, and Meteor framework.

LITERATURE STUDY

KNOWLEDGE MANAGEMENT SYSTEM

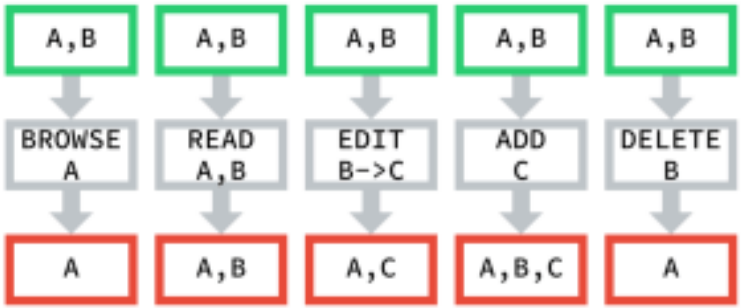
Data-Information-Knowledge-Wisdom (DIKW),
Knowledge Management System (KMS),
Personal Knowledge Manager

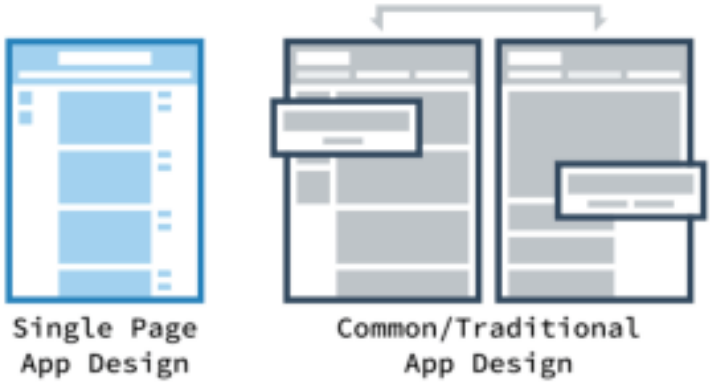
SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)



Agile methodologies,
Minimum Viable Product (MVP)

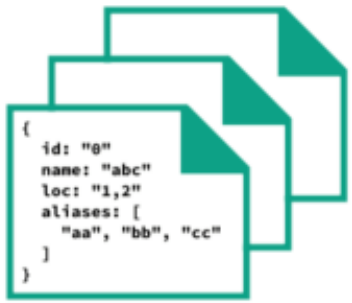
DESIGN PATTERN





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

DATABASE



Document/Schema-less
Data Model

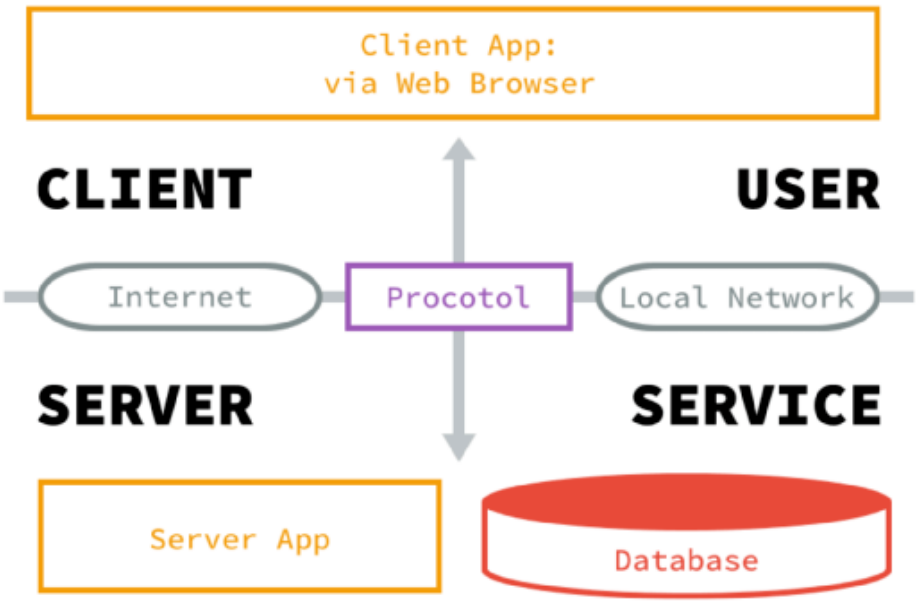
mongoDB

Document-based NoSQL database named MongoDB

With JavaScript and JSON related technologies

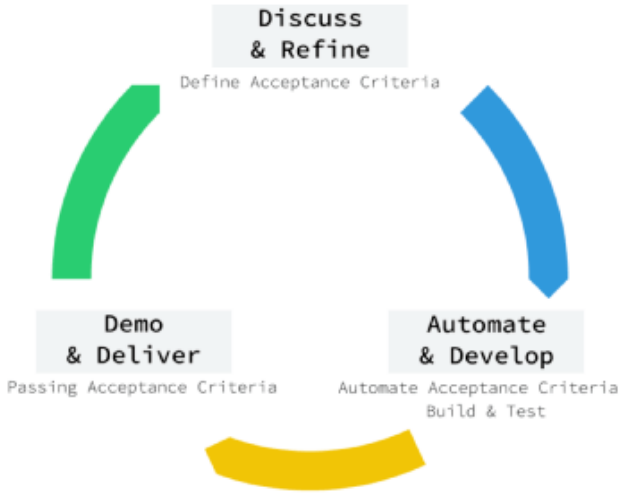
on top of Node.js

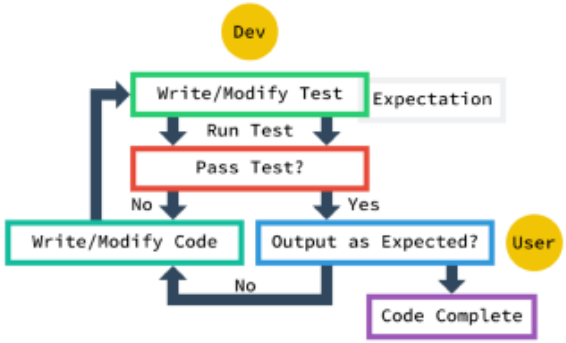
WEB APPLICATION



build a Web application

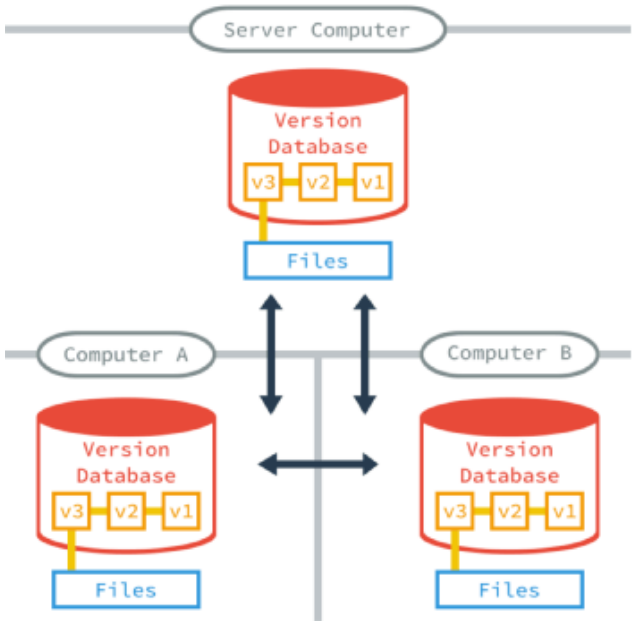
using full stack framework named Meteor

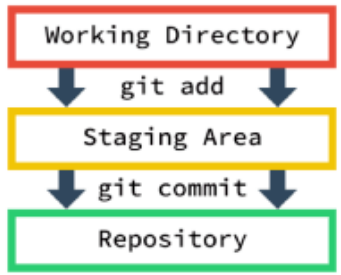




software tested in Acceptance Test Driven Development (ATDD)

SOURCE CODE MANAGEMENT





source code is version controlled using Git
and distributed via GitHub

ANALYSIS & DESIGN

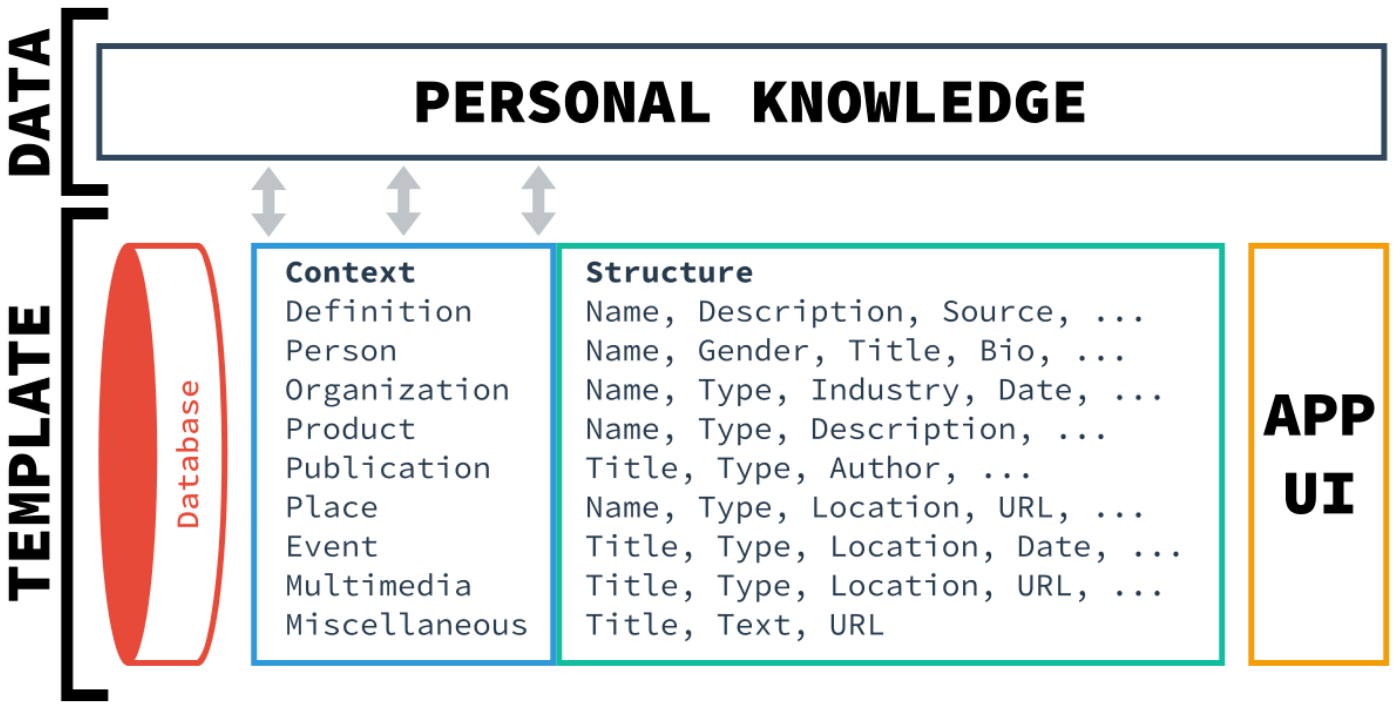
Target User:

Person who frequently gather and really want to manage their tons of knowledge at almost everytime

User Types:

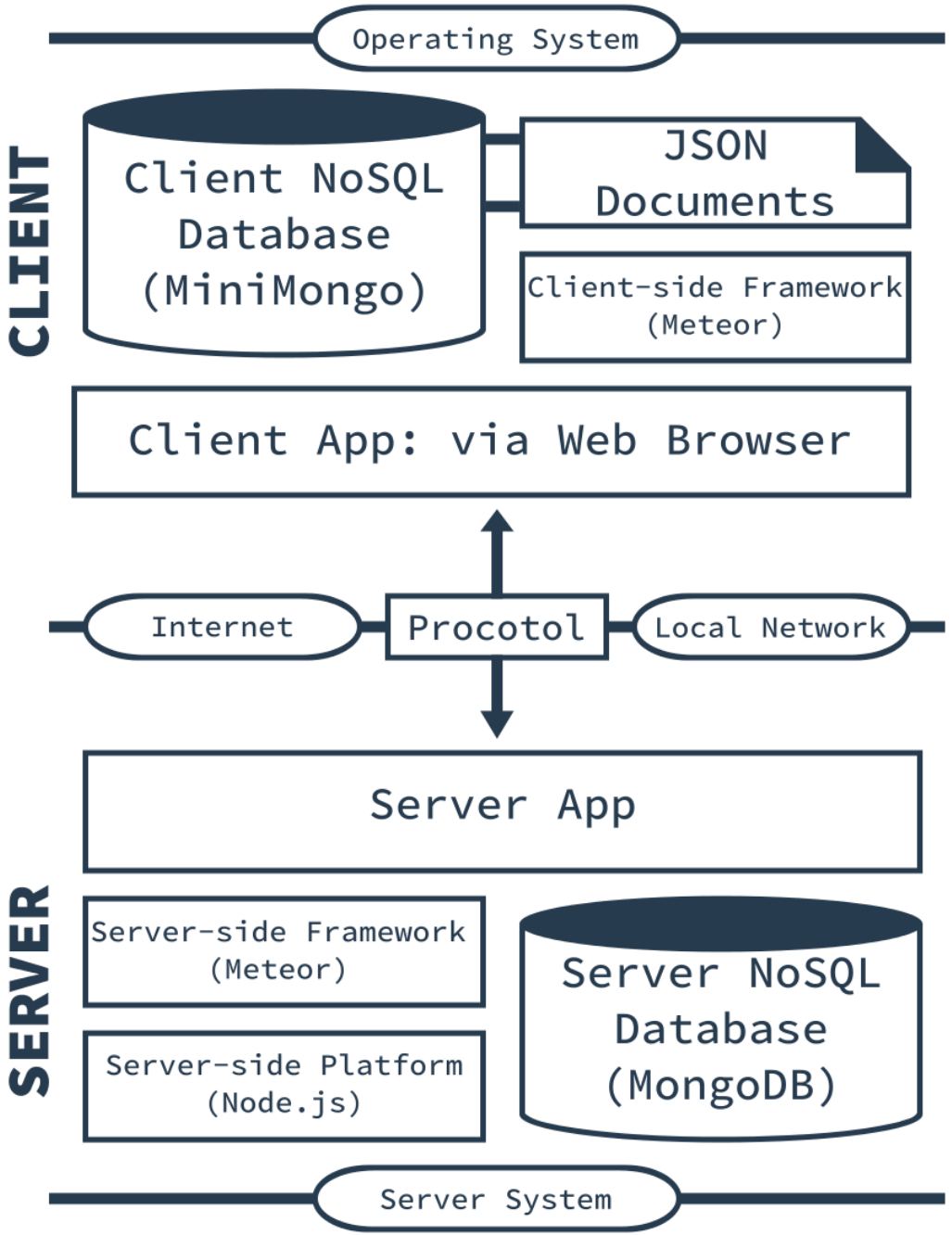
Regular person, researcher, engineer, developer, designer, information architect, event speaker, student, teacher, leader, collaborator, colleagues, recruiter, etc

DESIGN: CONTEXTUAL SYSTEM

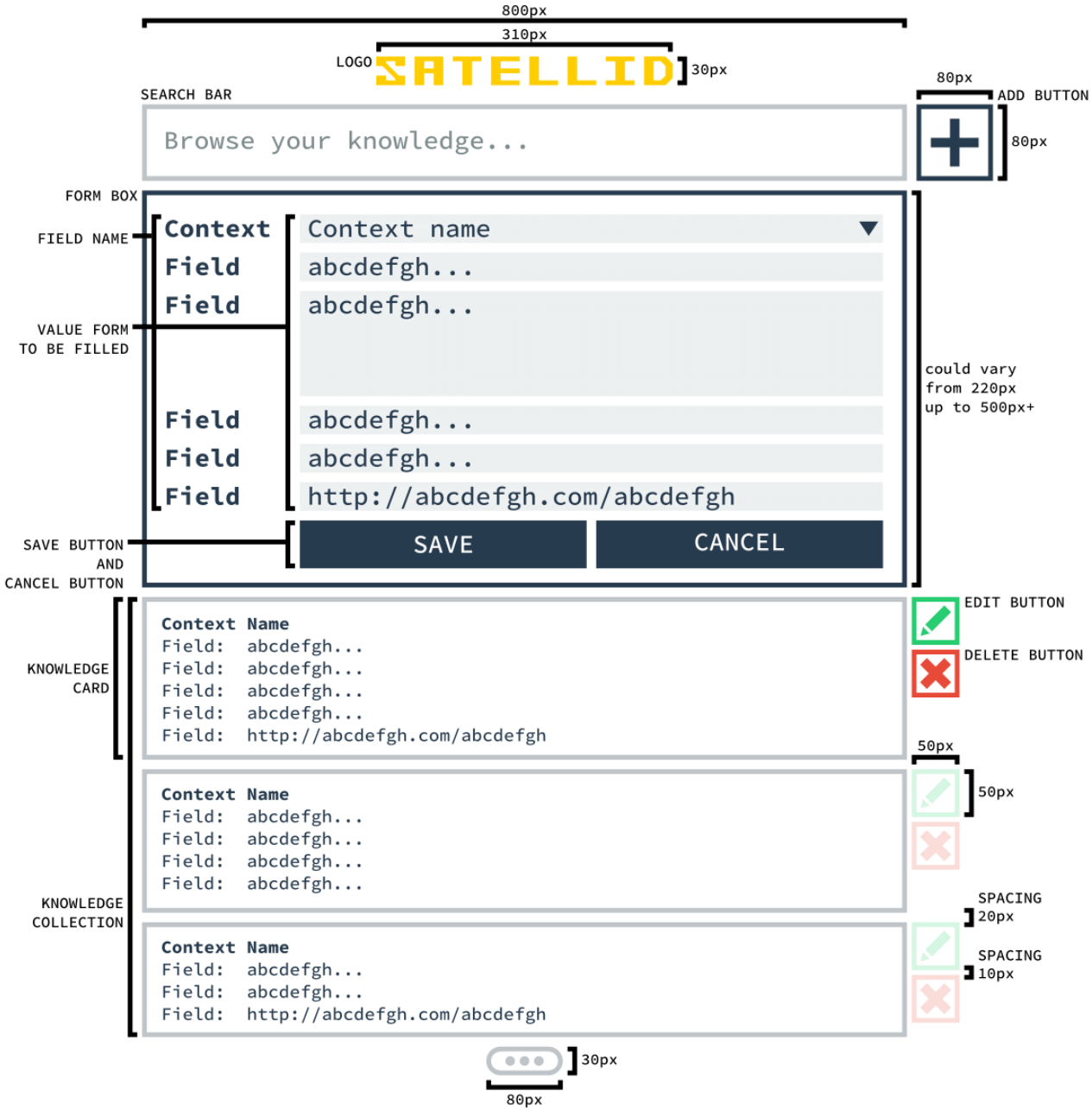


A data and template approach to classify the knowledge with its context and structure.

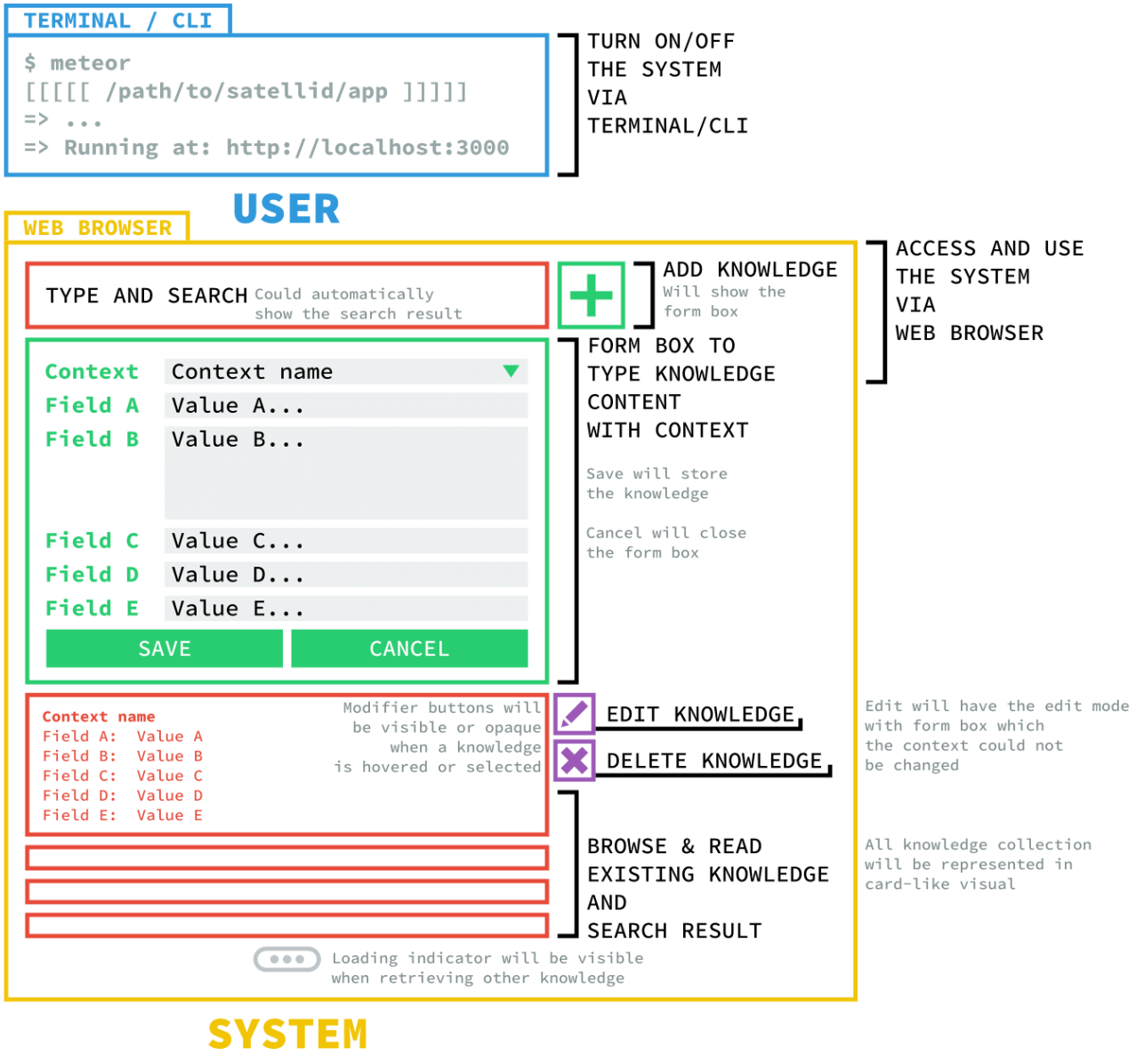
DESIGN: APPLICATION ARCHITECTURE



DESIGN: USER INTERFACE MOCKUP



DESIGN: USER INTERACTION



IMPLEMENTATION & TESTING

Snippets of Server Code

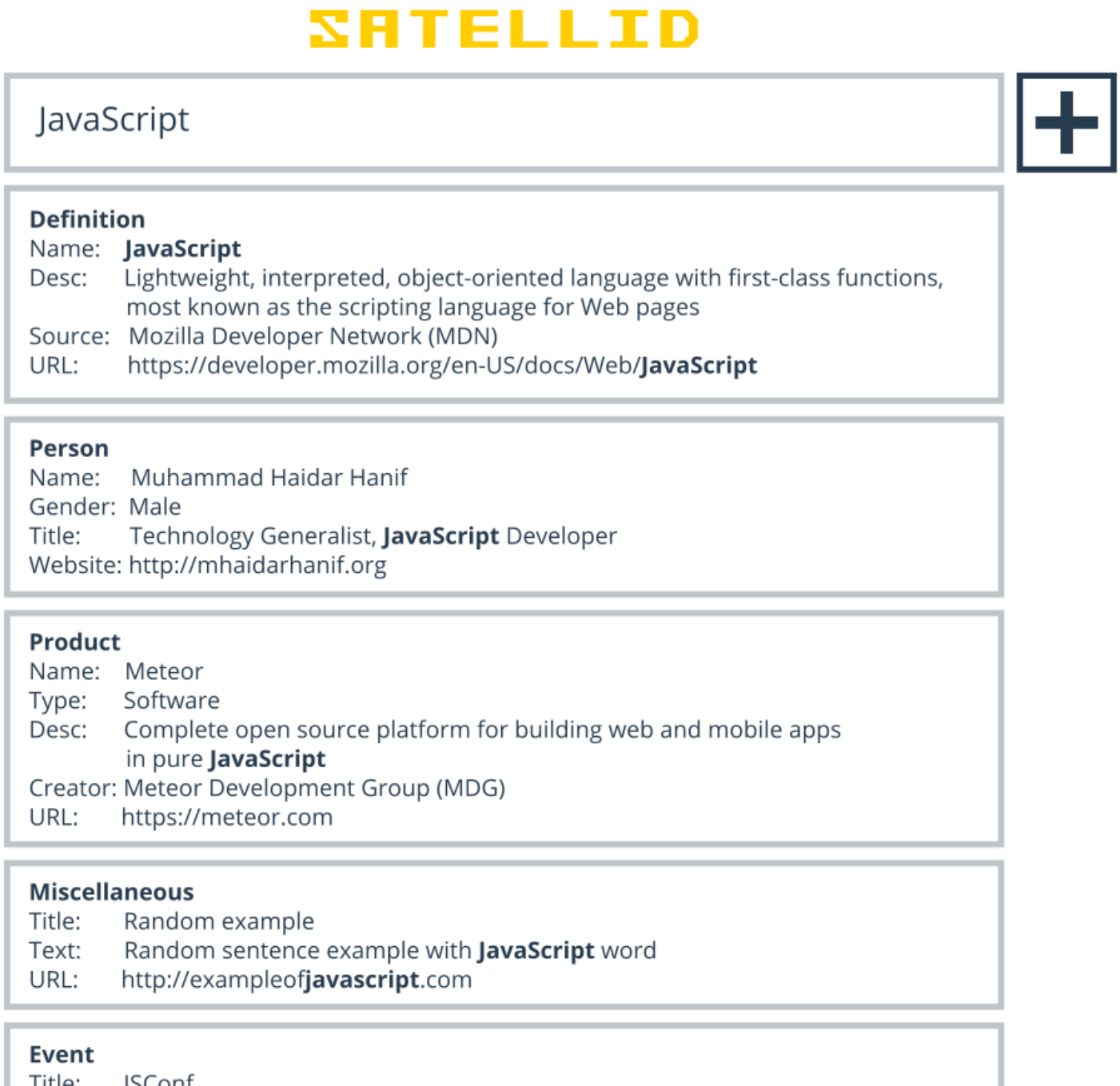
```
if (Meteor.isServer) {
  Meteor.startup(function() {
    // ...
  });
}
```

Snippets of Client Code

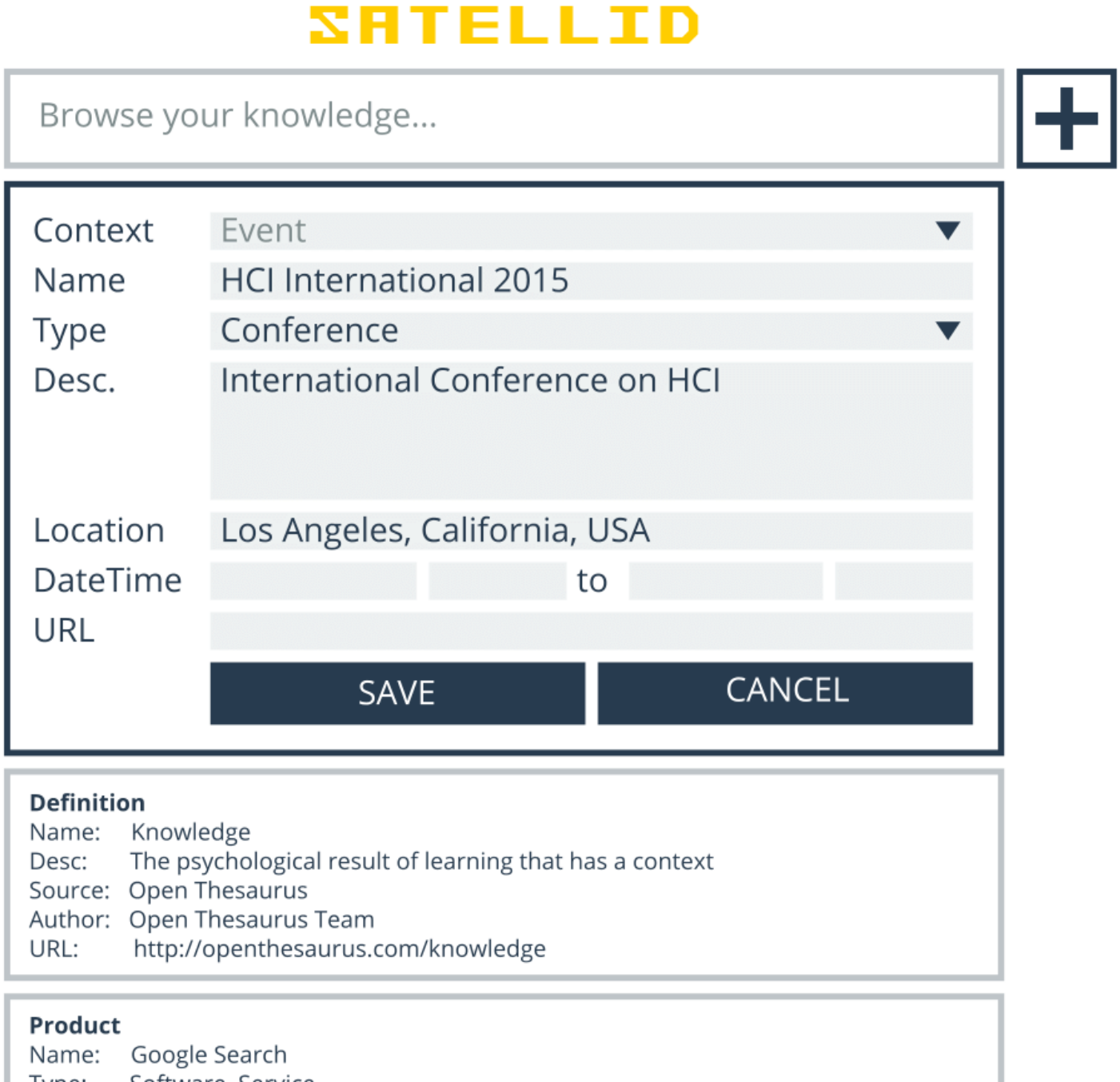
```
if (Meteor.isClient) {
  Template.add.events({
    "click": function() {
      // ...
    }
  });
}
```

APP RESULT SCREENSHOTS: READ

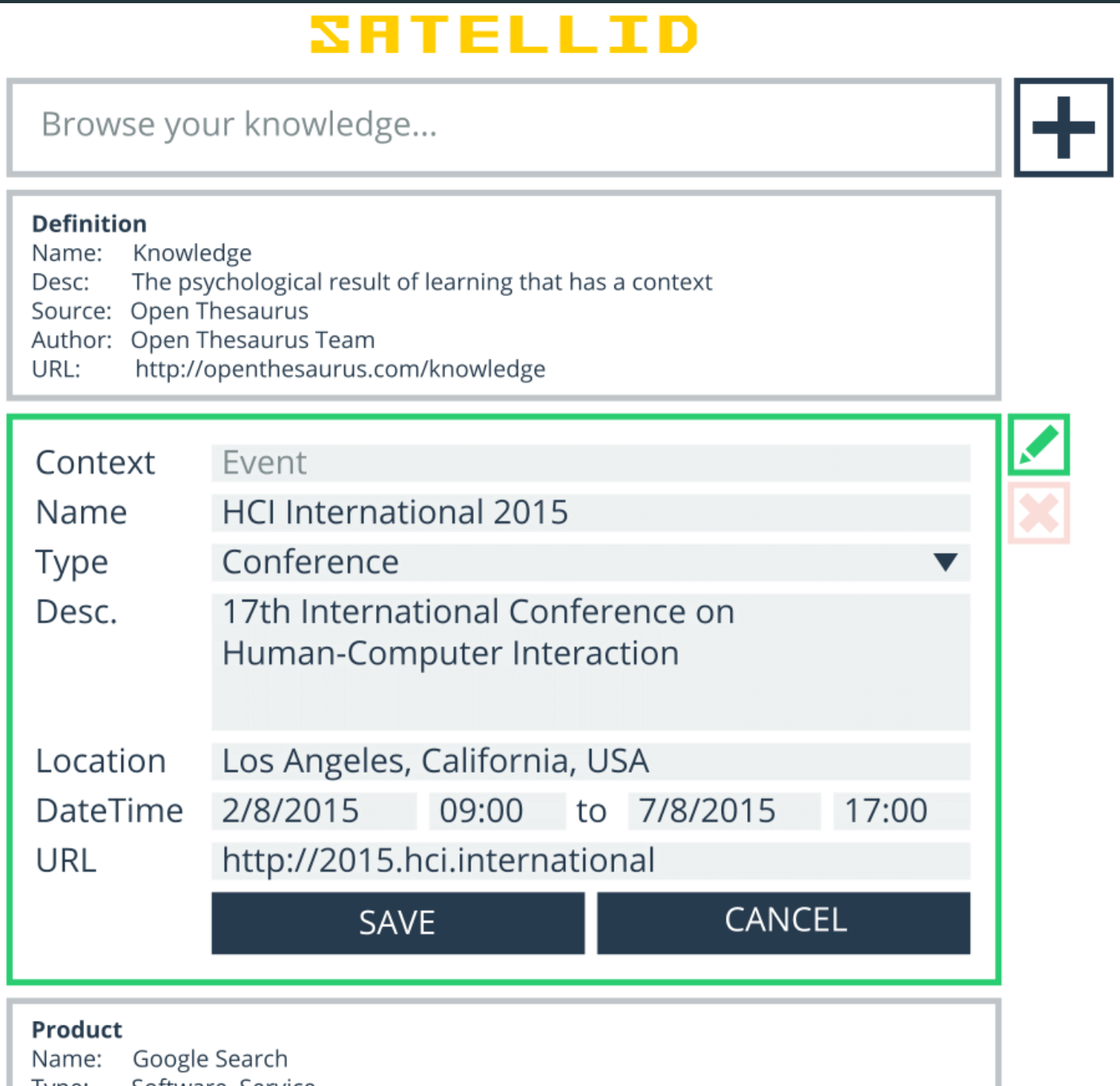
APP RESULT SCREENSHOTS: BROWSE



APP RESULT SCREENSHOTS: ADD



APP RESULT SCREENSHOTS: EDIT



APP RESULT SCREENSHOTS: DELETE

CLOSING

CONCLUSION

Simple and single system of managing tons of personal daily knowledge that implemented with Web technologies

SUGGESTION FOR FUTURE WORK

- Can have custom form
- Can easily import and export, including backup
- BREAD a template
- Multimedia support
- Integration with other networks
- Encryption

Thank You

Extra Notes

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 view 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

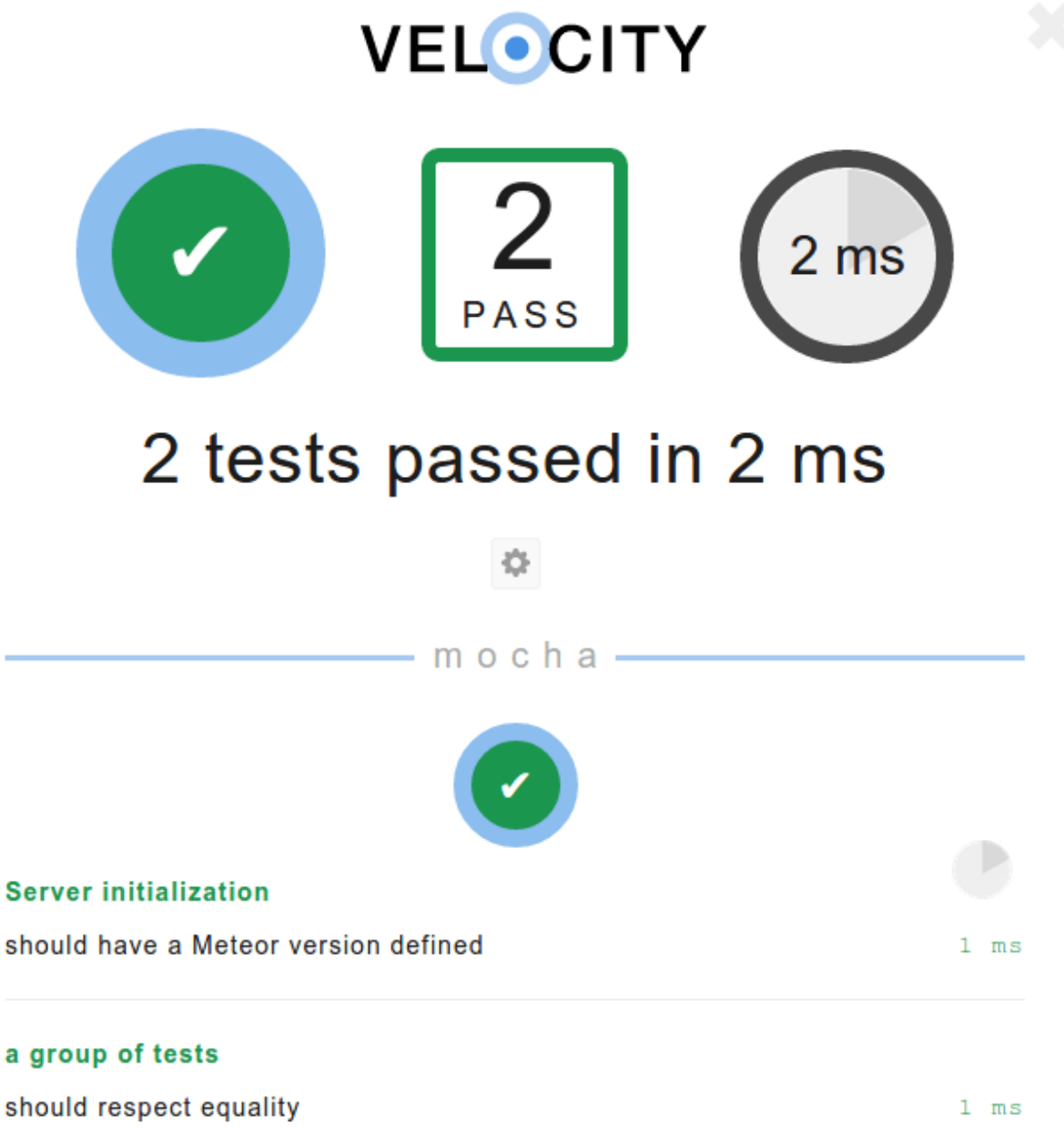
EXAMPLE OF FEATURE FILE

```
Feature: Example feature title

  As a user of this feature
  I want to have these steps
  So that I can have the expected result

  Scenario: Specific situation description
    Given a condition
    And an other condition
    When a performed action by someone
    Then an output is presented
```

TESTING RESULT WITH VELOCITY



SQL vs document-based (NoSQL)

id	name	loc
0	abc	1,2
1	def	3,4
2	ghi	5,6

Relational/Table Data Model

Document/Schema-less Data Model

- ✔ Represent document in real world
- ✔ Flexible schema
- ✔ Data nesting
- ✔ Various or complex data structure

Thank You