

Software Engineering
< DLMCSPSE01 />

Project: HoppyBrew

Concept Phase

International University of Applied Sciences

Written by
Asbjorn Bordoy

On the
18-Mar-2024

Revision 1.0

Abstract

The project aims to develop a comprehensive Beer Brewing Recipe Manager system, catering to brewing enthusiasts and homebrewers. This system facilitates the management of brewing processes and associated data through intuitive interfaces and robust functionalities. Users can create, share, and manage beer recipes, customize water and equipment profiles, schedule brewing sessions, monitor fermentation in real-time, generate reports, and more. The system ensures a seamless user experience by integrating with external devices like ISpindel for data collection and leveraging a database for secure storage and retrieval of brewing-related information. With an emphasis on user-friendly design and versatile features, the Beer Brewing Recipe Manager fosters innovation and tradition in the art of homebrewing.

Contents

1	Introduction and Goals	1
1.1	Quality Goals	1
1.2	Stakeholders	1
1.3	Requirements Overview	1
1.4	Business Context	5
1.5	Technical Context	5
1.6	Whitebox Overall System	5
1.6.1	<Name black box 1>	5
1.6.2	<Name black box 2>	5
1.6.3	<Name black box n>	5
1.6.4	<Name interface 1>	5
1.6.5	<Name interface m>	5
1.7	Level 2	5
1.7.1	White Box <building block 1>	5
1.7.2	White Box <building block 2>	5
1.7.3	White Box <building block m>	6
1.8	Level 3	6
1.8.1	White Box <_building block x.1_>	6
1.8.2	White Box <_building block x.2_>	6
1.8.3	White Box <_building block y.1_>	6
1.9	<Runtime Scenario 1>	6
1.10	<Runtime Scenario 2>	6
1.11	6
1.12	<Runtime Scenario n>	6
2	Deployment View	6
2.1	Infrastructure Level 1	6
2.2	Infrastructure Level 2	6
2.2.1	<Infrastructure Element 1>	6
2.2.2	<Infrastructure Element 2>	6
2.2.3	<Infrastructure Element n>	6
2.3	<Concept 1>	6
2.4	<Concept 2>	7
2.5	<Concept n>	7
2.6	Quality Tree	7
2.7	Quality Scenarios	7
3	Risks and Technical Debts	7
4	Glossary	7
	Bibliography	8

List of Tables

1	Quality goals and priorities for the application.	1
---	---	---

List of Figures

1	UseCases	2
2	sds	3
3	vignettes/test.png	4

1 Introduction and Goals

This document outlines the architecture and design of HoppyBrew, a web-based application for managing brewing recipes and brews. The application is designed to be user-friendly and intuitive, with a clean and modern user interface. The application is also designed to be compatible with a wide range of devices and browsers, and to integrate with other brewing tools and services, such as **iSpindel**.

Note! The terminology **brew** and **batch** are used interchangeably in this document to refer to the same thing, i.e. a single brewing process.

1.1 Quality Goals

The top three quality goals for the architecture and design whose fulfillment is of highest importance to the major stakeholders of HoppyBrew have been identified as follows:(Booch et al. 2007)

Table 1: Quality goals and priorities for the application.

Priority	Key word	Quality Goal
1	Usability	The application should be easy to use and intuitive, with a clean and modern user interface.
2	Compatibility	The application should be compatible with a wide range of devices and browsers. (mobile, desktop, tablet)
3	Integration	The application should integrate with other brewing tools and services, such as iSpindel .

The motivation behind these goals are to ensure that the application lives up to the expectations of the most important stakeholders, since they are the ones who will be the ones who influence the fundamental architecture and design decisions.

1.2 Stakeholders

In the architecture and design process of HoppyBrew, stakeholders play a pivotal role, providing essential requirements and constraints. Given that this project is part of a school assignment, the stakeholders are limited to the following individuals and their expectations:

Priority	Name/Category	Expectations
Primary	Beer Brewer Enthusiast	Wants a user-friendly and intuitive application for managing brewing recipes and brew logs.
Secondary	Self-hosting Enthusiast and developers	Wants a high-quality, open-source application that is easy to maintain and extend.

1.3 Requirements Overview

HoppyBrew is driven by the following essential features and functional requirements:

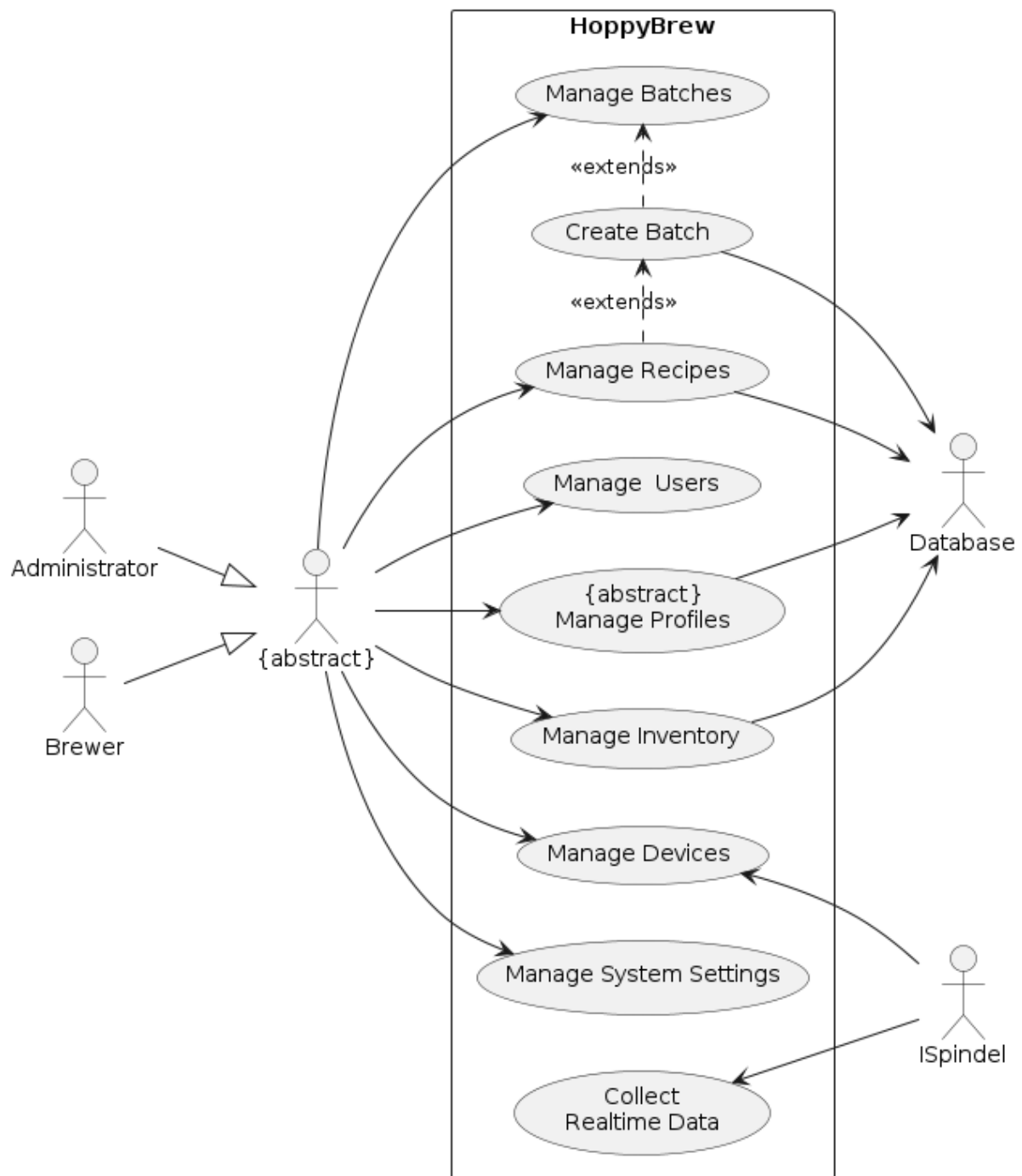


Figure 1: UseCases

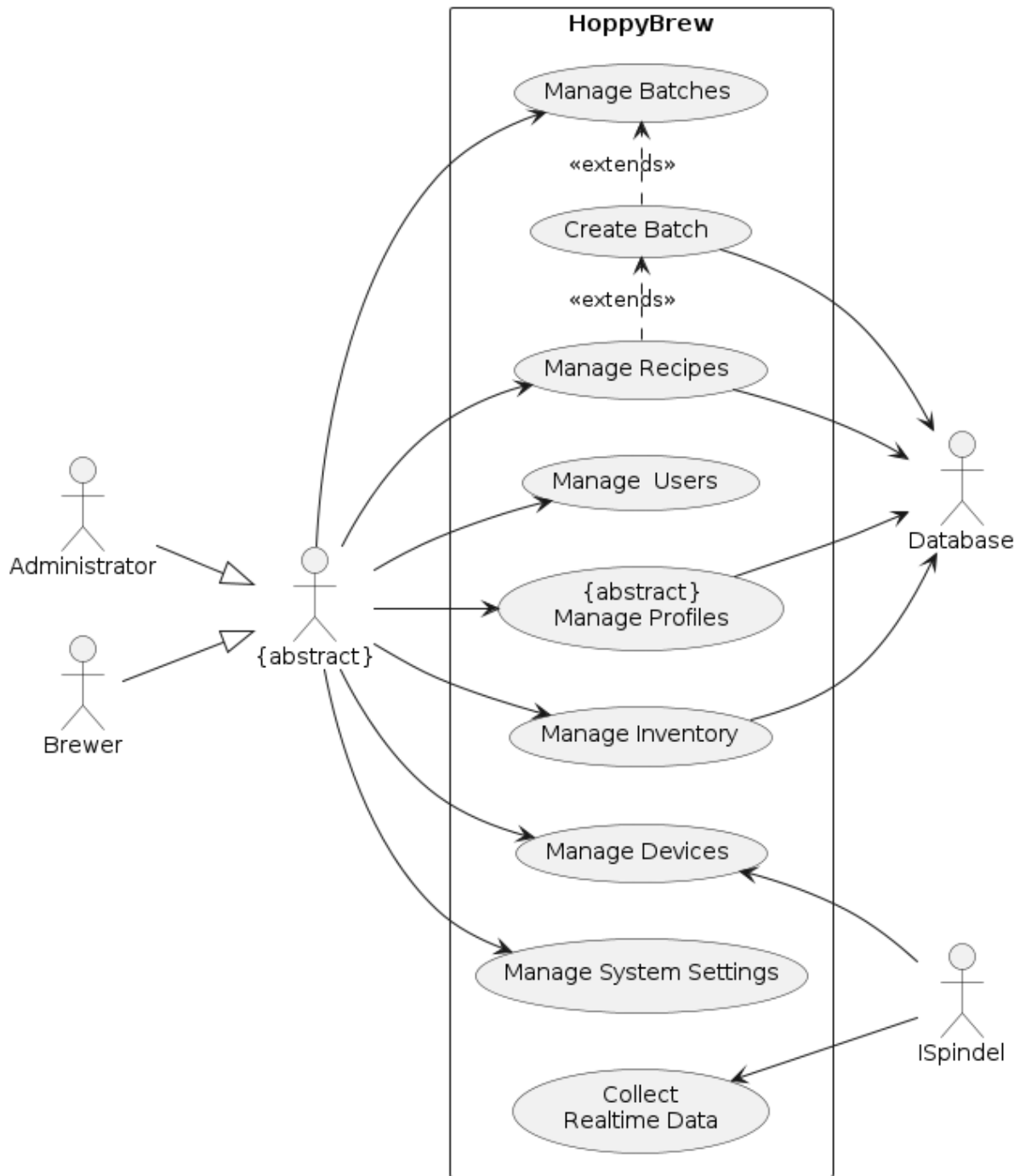


Figure 2: sds

Id	Requirement	Explanation
F1	Manage and create brewing recipes	The application should allow users to manage and create brewing recipes.
F2	Manage and create brews and log their progress	The application should allow users to manage and create brews and log their progress.

And here is the file

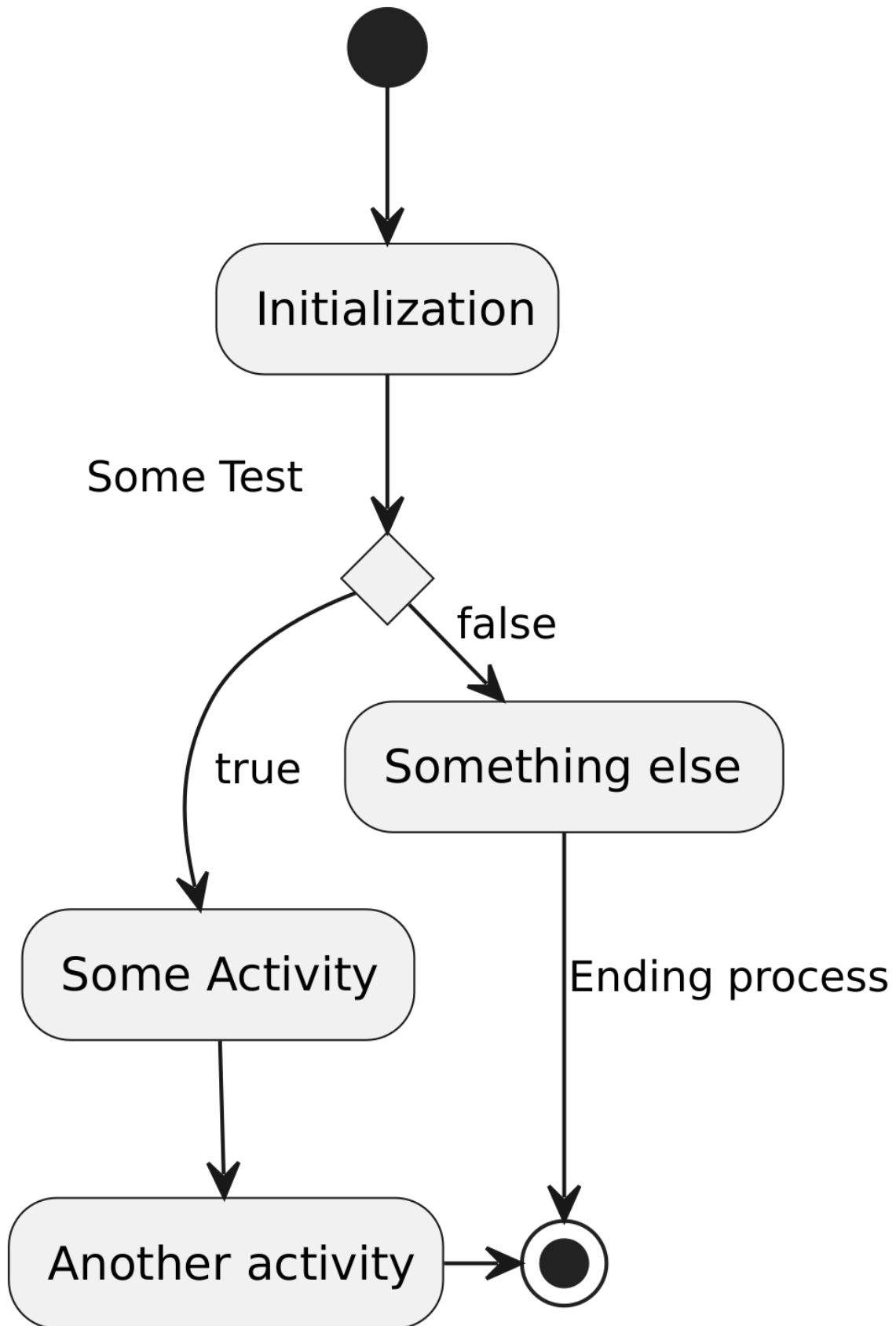


Figure 3: vignettes/test.png

(“Brewfather,” n.d.) # Architecture Constraints # System Scope and Context

1.4 Business Context

<Diagram or Table>

<optionally: Explanation of external domain interfaces>

1.5 Technical Context

<Diagram or Table>

<optionally: Explanation of technical interfaces>

<Mapping Input/Output to Channels> # Solution Strategy # Building Block View

1.6 Whitebox Overall System

<*Overview Diagram*>

Motivation

<*text explanation*>

Contained Building Blocks

<*Description of contained building block (black boxes)*>

Important Interfaces

<*Description of important interfaces*>

1.6.1 <Name black box 1>

<*Purpose/Responsibility*>

<*Interface(s)*>

<(Optional) *Quality/Performance Characteristics*>

<(Optional) *Directory/File Location*>

<(Optional) *Fulfilled Requirements*>

<(optional) *Open Issues/Problems/Risks*>

1.6.2 <Name black box 2>

<*black box template*>

1.6.3 <Name black box n>

<*black box template*>

1.6.4 <Name interface 1>

...

1.6.5 <Name interface m>

1.7 Level 2

1.7.1 White Box <*building block 1*>

<*white box template*>

1.7.2 White Box <*building block 2*>

<*white box template*>

...

1.7.3 White Box <building block m>

<white box template>

1.8 Level 3

1.8.1 White Box <__building block x.1__>

<white box template>

1.8.2 White Box <__building block x.2__>

<white box template>

1.8.3 White Box <__building block y.1__>

<white box template> # Runtime View

1.9 <Runtime Scenario 1>

- <insert runtime diagram or textual description of the scenario>
- <insert description of the notable aspects of the interactions between the building block instances depicted in this diagram.>

1.10 <Runtime Scenario 2>

1.11 ...

1.12 <Runtime Scenario n>

2 Deployment View

2.1 Infrastructure Level 1

<Overview Diagram>

Motivation

<explanation in text form>

Quality and/or Performance Features

<explanation in text form>

Mapping of Building Blocks to Infrastructure

<description of the mapping>

2.2 Infrastructure Level 2

2.2.1 <Infrastructure Element 1>

<diagram + explanation>

2.2.2 <Infrastructure Element 2>

<diagram + explanation>

...

2.2.3 <Infrastructure Element n>

<diagram + explanation> # Cross-cutting Concepts

2.3 <Concept 1>

<explanation>

2.4 <Concept 2>

<explanation>

...

2.5 <Concept n>

<explanation> # Architecture Decisions # Quality Requirements

2.6 Quality Tree

2.7 Quality Scenarios

3 Risks and Technical Debts

4 Glossary

Term	Definition
<Term-1>	<definition-1>
<Term-2>	<definition-2>

Bibliography

Booch, Grady, Robert Maksimchuk, Michael Engle, Bobbi Young, Jim Conallen, and Kelli Houston. 2007. *Object-Oriented Analysis and Design with Applications, Third Edition*. Third. Addison-Wesley Professional.

“Brewfather.” n.d. Brewfather. <https://web.brewfather.app/>.