Software Architecture Document

Wanderlust

Jonathas Felipe da Silva

22/10/2019

Revision History

Approval Block

Table of Contents

Software Architecture Document

1.   Introduction

This document provides a high level overview and explains the architecture of the Wanderlust system.

The document defines goals of the architecture, the use cases supported by the system, architectural styles and components that have been selected. The document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.

1.1.   Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the Wanderlust system. It presents a number of different architectural views to depict the different aspects of the system.

In order to depict the software as accurately as possible, the structure of this document is based on Philippe Kruchten's "4+1" model view of architecture [Kruchten].
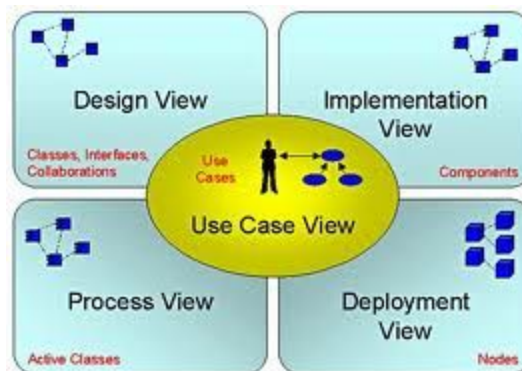
The "4+1" View Model allows various stakeholders to find what they need in the software architecture.

1.2.   Scope

The scope of this SAD is to explain the architecture of the Distributed Development Monitoring and Mining system.

This document describes the various aspects of the Wanderlust system design that are considered to be architecturally significant. These elements and behaviors are fundamental for guiding the construction of the Wanderlust system and for understanding this project as a whole. Stakeholders who require a technical understanding of the Wanderlust system are encouraged to start by reading the Project Proposal, Concept of Operations and Software Requirements Specification documents developed for this system [PP, ConOps, SRS].

1.3.   Definitions, Acronyms, and Abbreviations

- **SAD -** Software Architecture Document
- **UML** – Unified Modeling Language
- **User -** This is any user who is registered on the Wanderlust app

1.4.   References

[PP]:   Project Proposal

[SPMP]:   Software Project Management Plan

[ConOps]:   Concept of Operations

[SRS]:   Software Requirements Specification

[MedBiquitous]: Sample SAD, http://medbiq.org/std_specs/techguidelines/softwarearchitecture.pdf

[Kruchten]: The "4+1" view model of software architecture, Philippe Kruchten, November 1995, http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf

1.5.   Overview

In order to fully document all the aspects of the architecture, the Software Architecture Document contains the following subsections.

Section 2: describes the use of each view

Section 3: describes the architectural goals and constraints of the system

Section 4: describes the most important use-case realizations

2.   Architectural Representation

This document details the architecture using the views defined in the "4+1" model [Kruchten]. The views used to document the Wanderlust system are:

Use Case view
Audience: all the stakeholders of the system, including the end-users.
Area: describes the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describes the actors and use cases for the system, this view presents the needs of the user and is elaborated further at the design level to describe

discrete flows and constraints in more detail. This domain vocabulary is independent of any processing model or representational syntax (i.e. XML).
Related Artifacts : Use-Case Model, Use-Case documents

3.   Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. The system is meant as a proof of concept for a more complete project prediction system to be built in the future. Therefore one of the primary stakeholders in this document and the system as a whole are future architects and designers, not necessarily users as is normally the case. As a result, one goal of this document is to be useful to future architects and designers.

2. The Wanderlust app will be developed for the Android system through the Android Studio IDE and will use the SQLite library for data persistence.

3. The system will communicate with the Dialogflow API to build chatbot Wand, which is responsible for passing recommendations to users.

4.   Use-Case View

The purpose of use case visualization is to provide additional context around system usage and the interactions between its components. For the purposes of this document, each component is considered a use case actor. Section 4.1 lists the current actors and provides a brief description of each in the general context of system use. In section 4.2, the use case diagrams, activity and their sequence for the application are shown.

4.1 Actors

4.1.1 - a User

Main actor, agent who will perform the functions available in the system. The user will have a login and password, elements that will allow access to available features, such as login, password, typical queries and interaction with the BOT.

4.1.2 - a BOT

Part of the system, the bot linked to artificial intelligence, will be responsible for providing the user with information related to local customs and traditions.

4.2 Use-Case Realizations

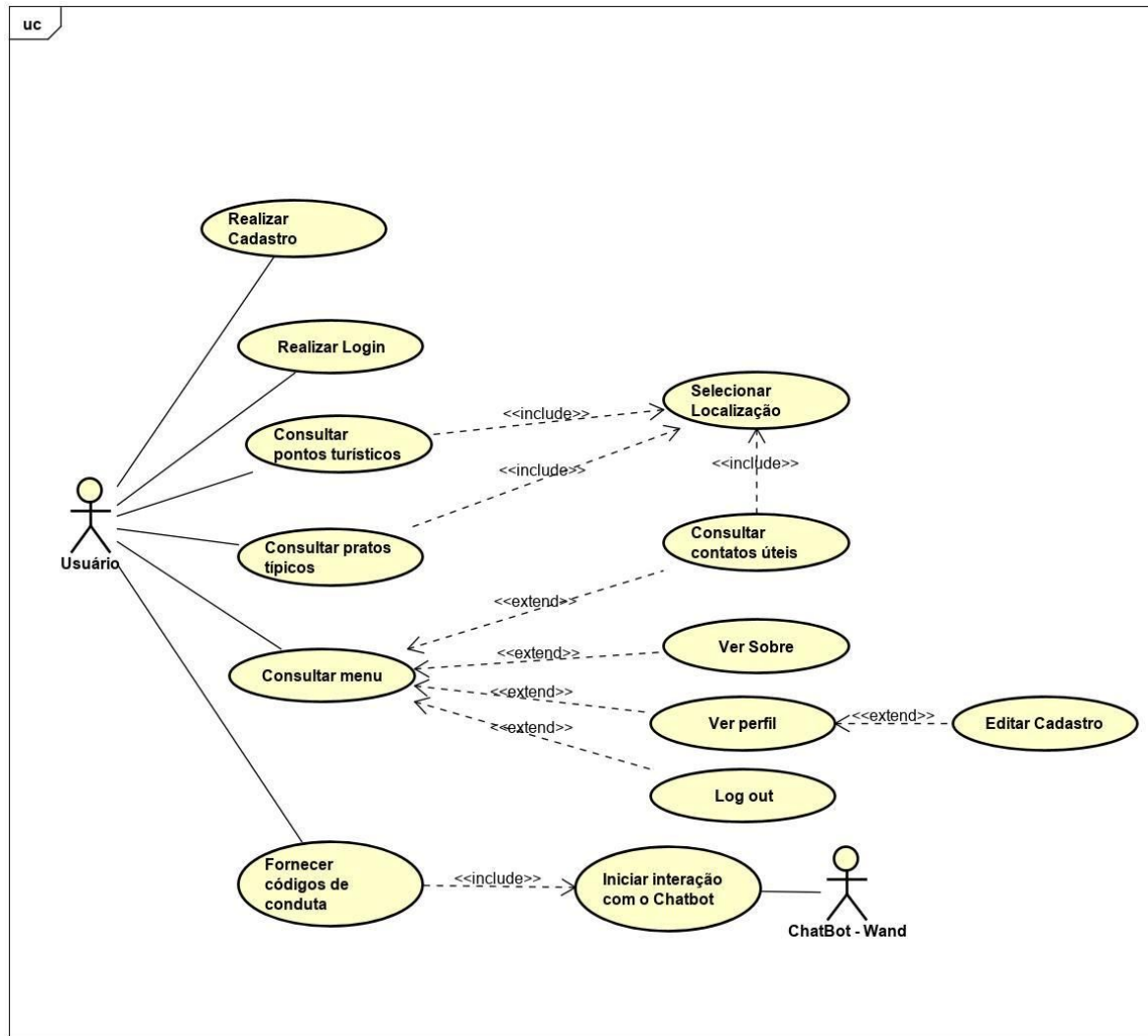**Figure 4.2.1** Wanderlust Use Case Diagram
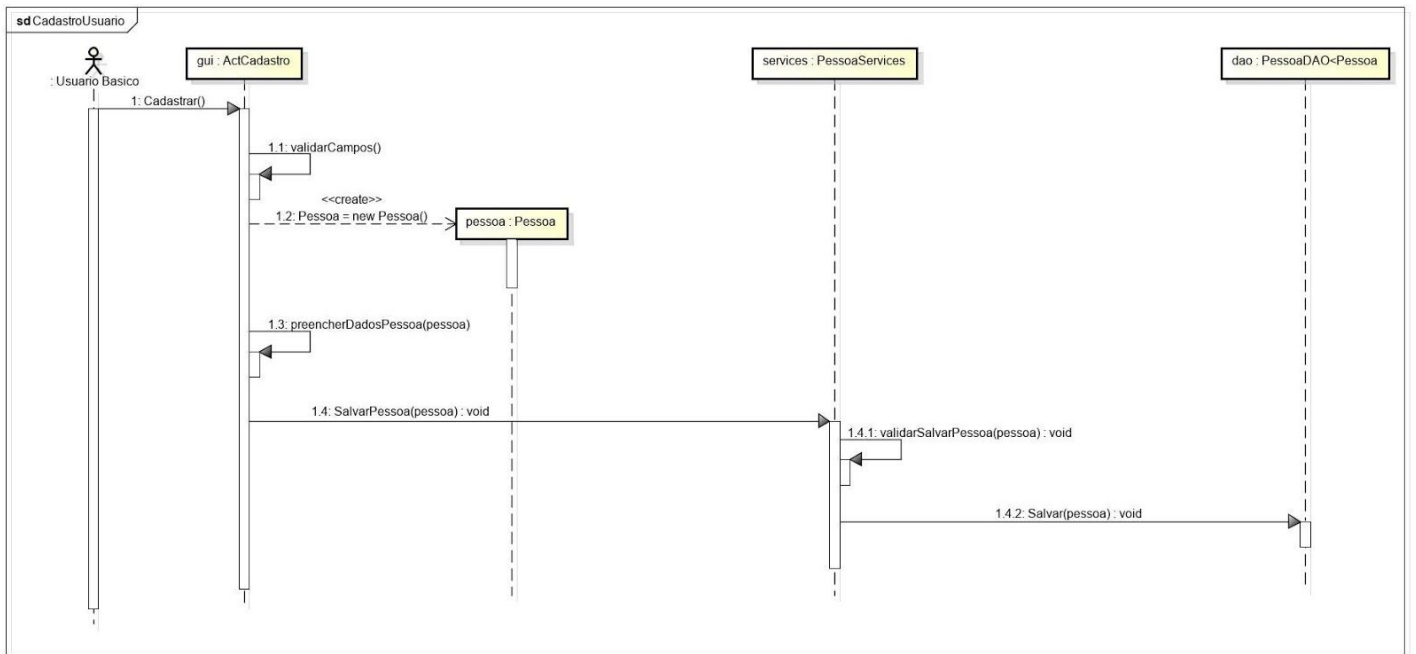
Figure 4.2.2 Register Sequence Diagram

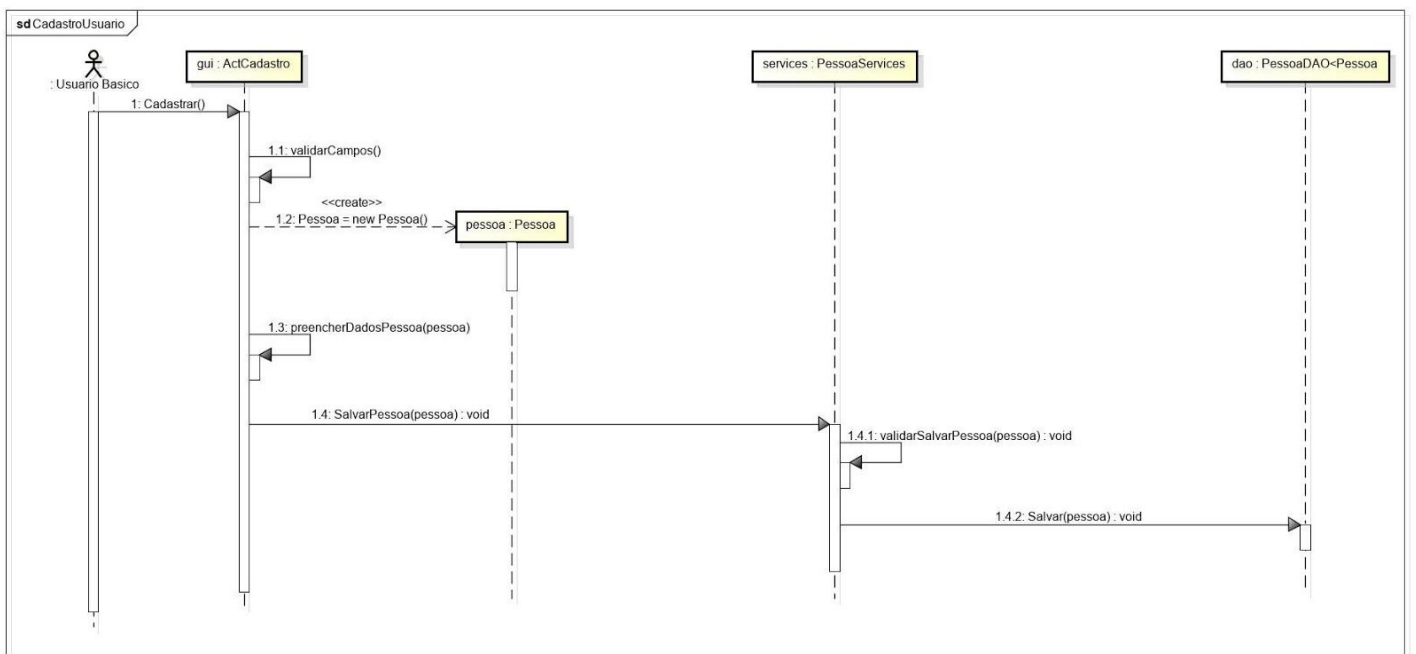
Figure 4.2.3 Login Sequence Diagram
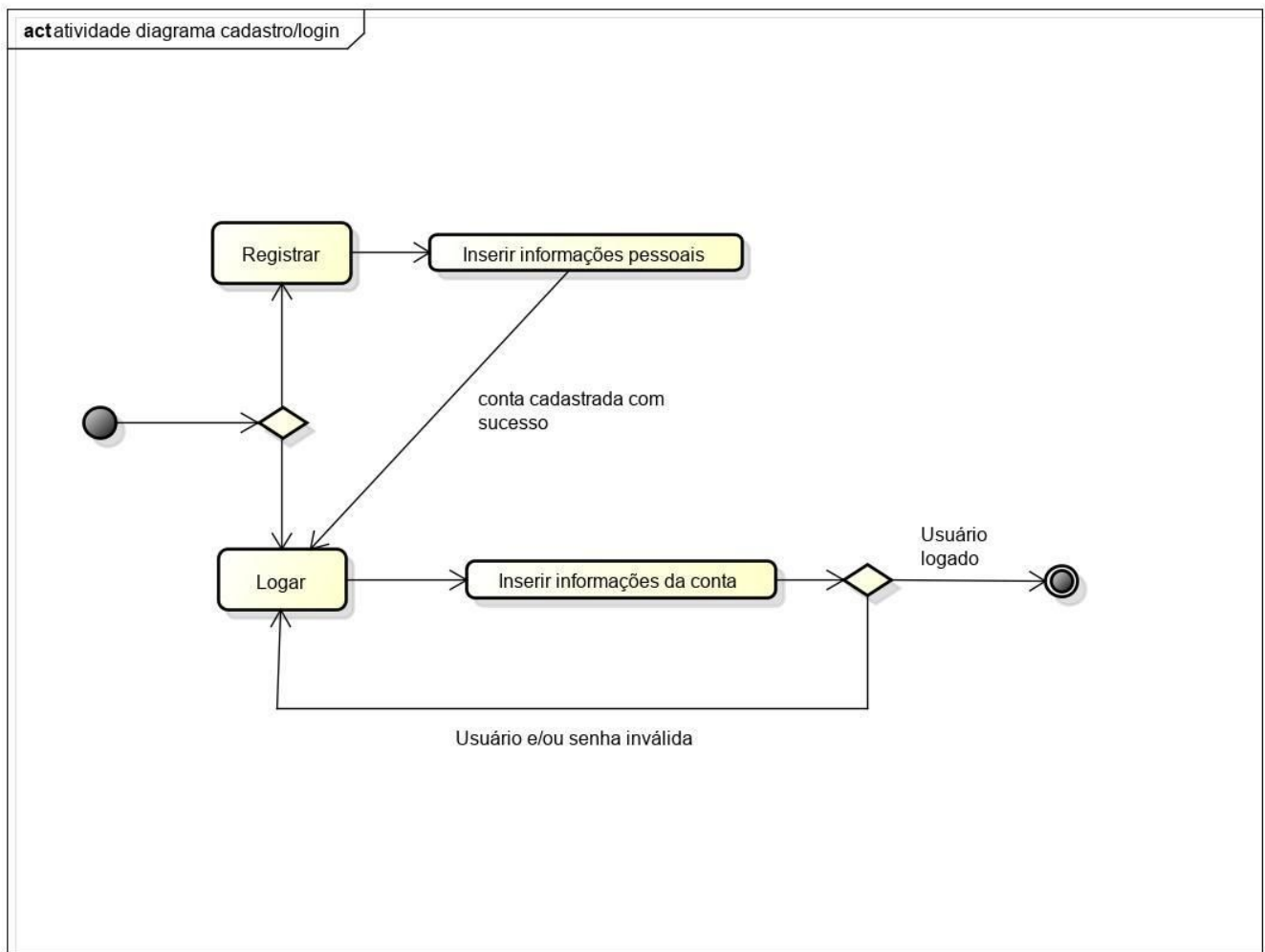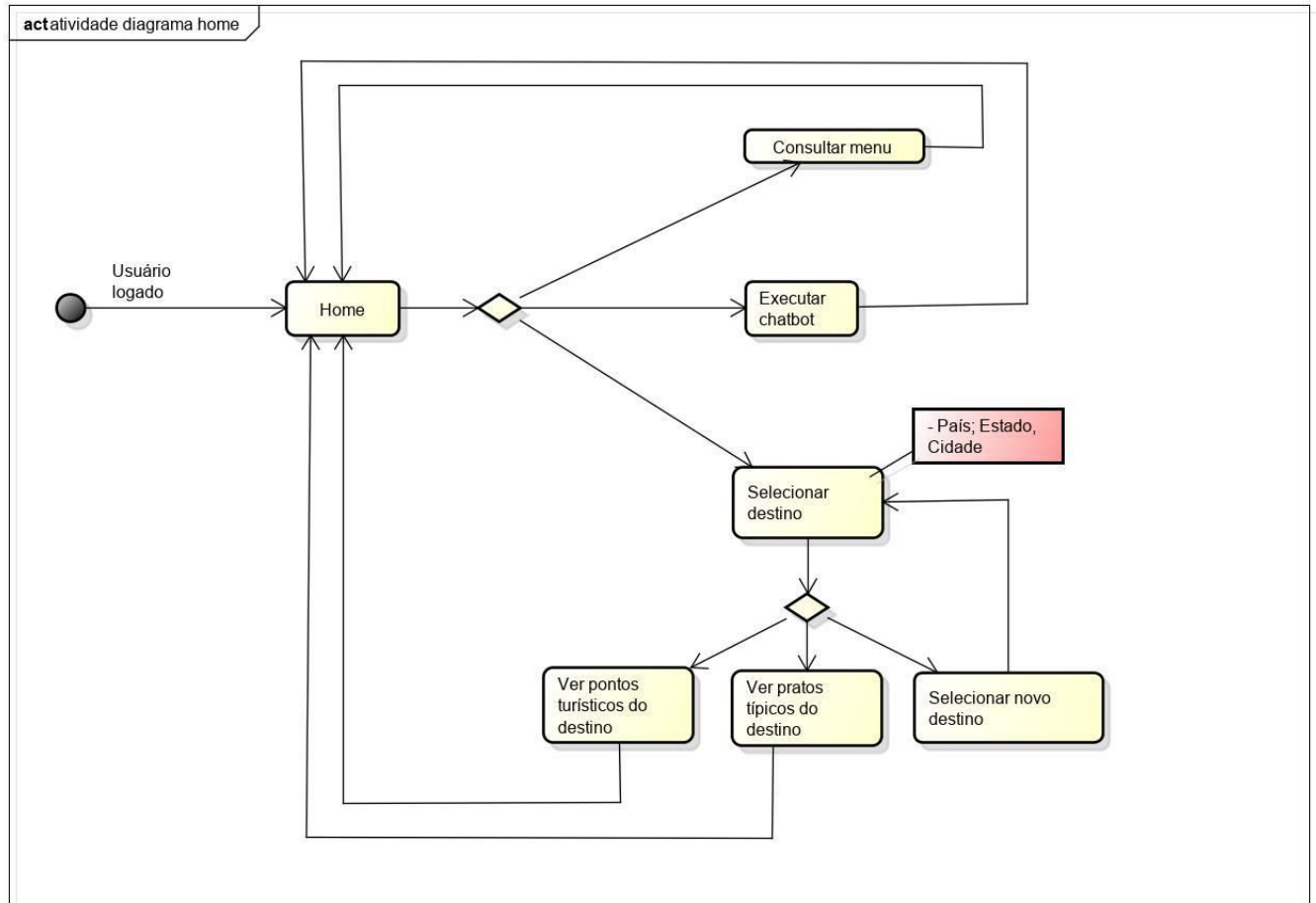
Figure 4.2.4 Login/Register Activity Diagram

Figure 4.2.5 Home Activity Diagram

Figure 4.2.6 Menu Activity Diagram