Znailla Claims System Solution Architecture

**Version 1.0**

**07/September/2019**

**Document History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Description** | **Submitted by** |
| 1.0 | Initial Draft | Sameh Amin |

**Table of Contents**

[Functional View 2](#__RefHeading___Toc110_3050861345)

[Functional Flow / Use Cases 2](#__RefHeading___Toc122_3050861345)

[Context Diagram 2](#__RefHeading___Toc124_3050861345)

[Quality Attributes 4](#__RefHeading___Toc114_3050861345)

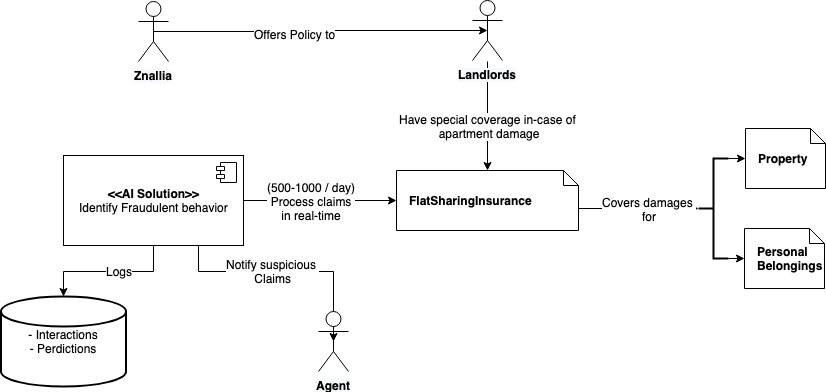
[Security perspective 4](#__RefHeading___Toc116_3050861345)

[Concurrency perspective 4](#__RefHeading___Toc118_3050861345)

[Performance perspective 4](#__RefHeading___Toc120_3050861345)

# Functional View

## Context Diagram

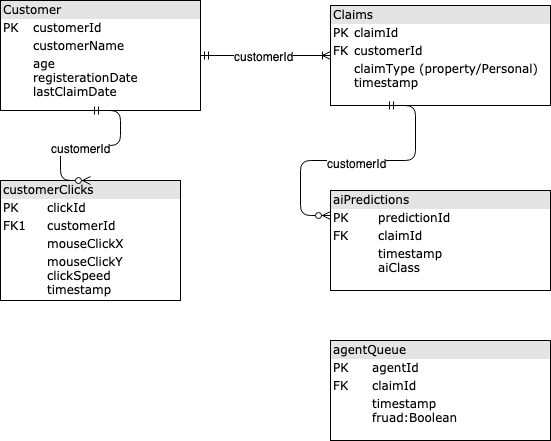
****

## Functional Flow / Use Cases

|  |  |  |
| --- | --- | --- |
| **ID** | **Use Case** | **Actor** |
| 1 | Submit claims through Znailla website | Customer |
| 2 | AI Service to predict and classify fraud claims | Znailla AI System |
| 3 | AI system to Log all interactions and prediction results | Znailla AI system |
| 4 | AI system to notify Agents about fraud claims | Znailla AI system |

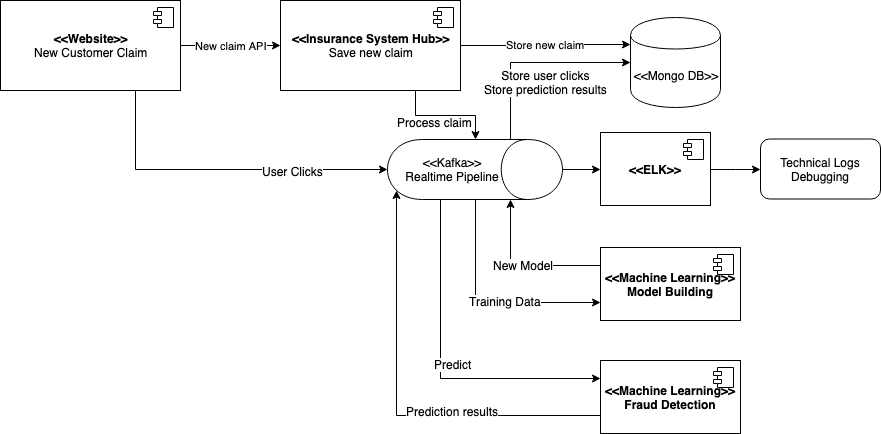
# Information View

## ERD



* This is an ERD to show all Entities suggested to store in the system.
* We’re going to use MongoDB for flexible schema, indexing and fast queries.
* Store basic information about the system such as
  + Claims
  + Customers
* Machine learning Features
  + User clicks on the insurance website should be stored In order to use as an important feature by the anomaly detection machine learning algorithm.
  + Customer information also could be important key features to use by the machine learning algorithm such as
    - Customer age
    - Registration date
    - Last claim date
  + Claims data also can be used as machine learning features, such as claimType.
* Also we need to store all AI Predictions related to all claims and the class matched with this claim, in this case it’s binary classification (either Fraud or not). I name it aiClass so we’re able to expand and add more classes in the future.

## DataFlow



# Deployment View

# Operational View

# Quality Attributes

## Security perspective

TBD

## Availability perspective

TBD

## Performance perspective

TBD