Instrument Recognition Software: Architecture and Design Specifications

Aleks Dziewulska - Aleksandra.Dziewulska@student.csulb.edu

Aner Gabriel - Aner.Gabriel@student.csulb.edu

Axel Valdez - Axel.CardenasValdez@student.csulb.edu

Cuong Pham - Cuong.Pham@student.csulb.edu

Joe Freedman - Joseph.Freedman@student.csulb.edu

Thomas Vugia - KyThuyen.Vugia@student.csulb.edu

CONTENTS

Architecture Design

Model:

Python is used to handle machine learning computations. Login credentials such as username and password are stored on a MongoDB database.

View:

JavaFX, Java's cross-platform graphical user interface API, is used to allow the user to interact with the algorithm with ease.

Controller:

A Python file will receive user updates sent by the view and trigger appropriate responses in the model. It also updates the view based on the response received from the model.

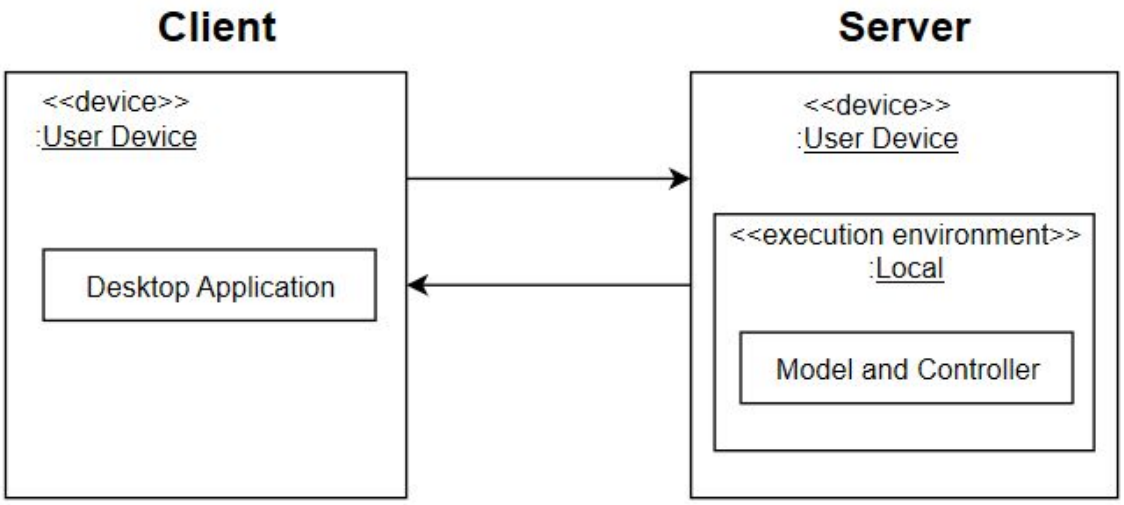
Client:

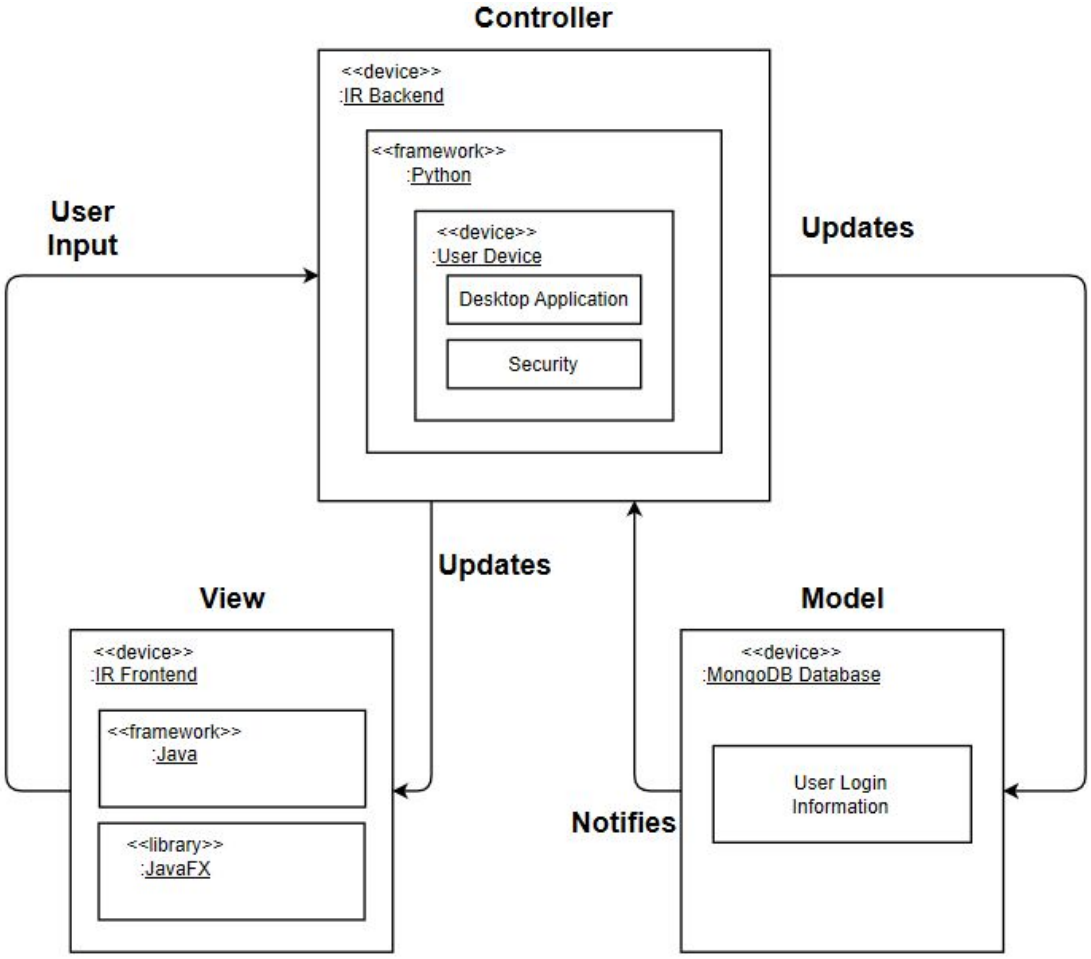
Cross-platform desktop application.

Server:

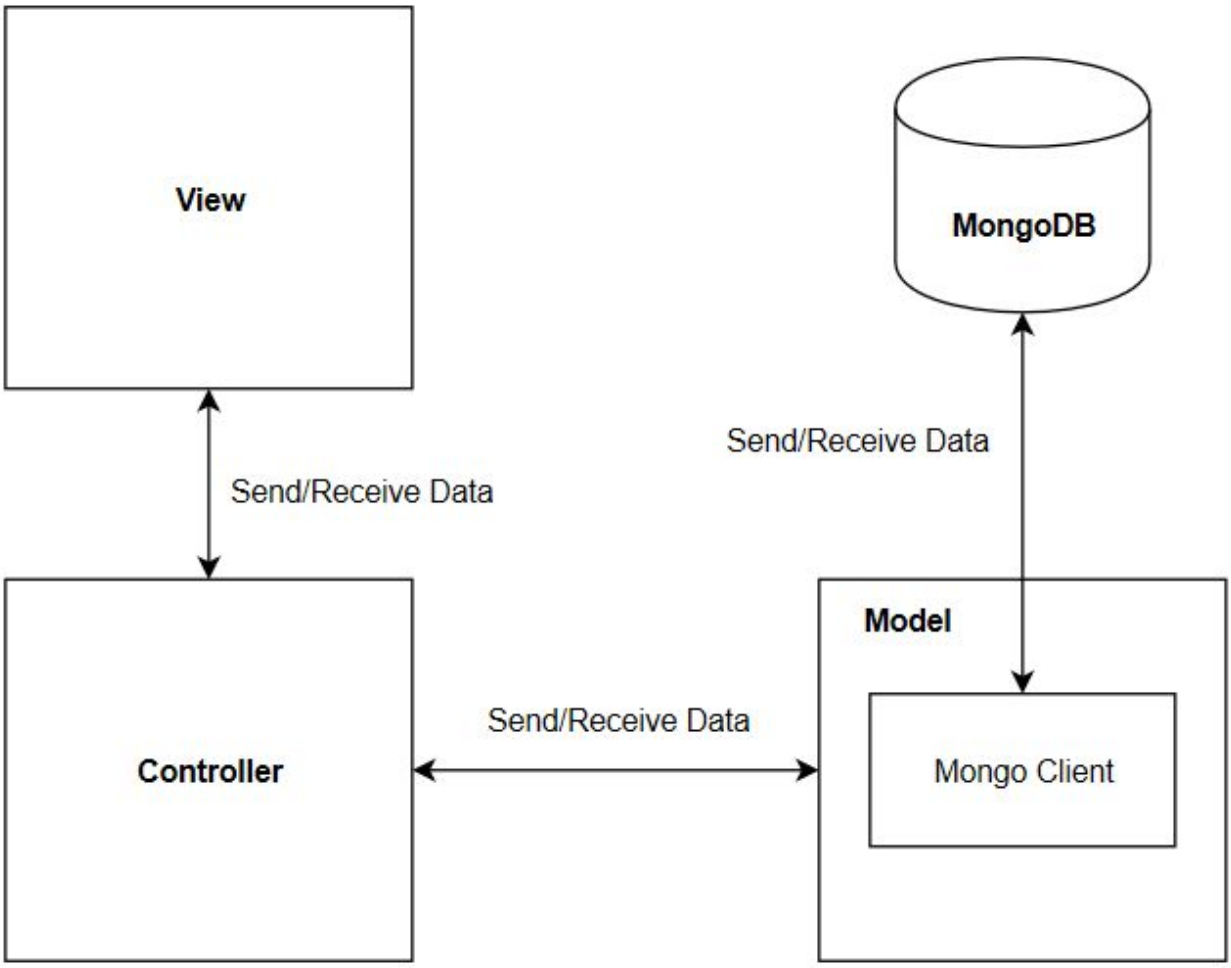
Hosted locally on user's device.

Component Design

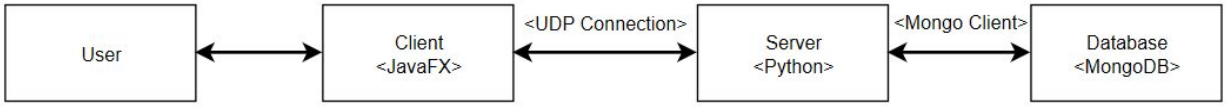




Database Connectivity



High Level Design



- User interacts with the Client.
- Client is connected to the Server using a local UDP Connection.
- Server uses a Mongo Client to connect to the Database.

Tradeoff Analysis

Architecture patterns/Language choice:

Criterion	Weight	Python	Java
Object Oriented	25%	0	1
Data Management	50%	1	0
Ease of Use	10%	1	0
Libraries	15%	1	0

Framework choice:

Criterion	Weight	Desktop	Web	Mobile
Data Management	70%	1	0	0
Computation Power	15%	1	1	0
Accessibility	5%	0	1	1
Server Reliability	10%	1	0	0

Database Choice:

Criterion	Weight	MongoDB	MySQL
High Availability	15%	1	0
Data Locality	60%	1	0
Auto-sharding	20%	1	0
Complex Transactions	5%	0	1