RoofUs Documentation

Lawrence Gathogo, Joseph Brannon, Aryan Yadav, Abdoul Samad Djido

Fall 2022

1 Project Definition

Brief Description A data science tool developed to assist in home pricing via machine learning. The information is accessed via website, and reported via web interface.

- **Why**
  - The purpose behind RoofUs is to better assist consumers on decision making in real estate. There is a disconnect between consumer and retailer and we want to fix that issue. This tool would help consumers understand the house pricing in a certain location, specifically North Carolina based on our dataset.
- **What**
  - Taking housing data and using machine learning and gradient boosting to predict prices for a home market. The end result will be accessible via API and maintained on our back-end. Others may be able to use our API if they choose to. Documentation for the API will be provided.
- **How**
  - Python will be used for generating a machine learning model to predict housing cost. A website will be used to demonstrate the results of the model, and to intake basic user inputs. Potentially a database to manage and store basic queries or data from users.

2 Project Requirements

- Functional
  1. Inputs: Zip code, bed and bath size
  2. Compares user input with national average values via graph/chart
- Usability
  1. User interface- The website we will create for the user to enter specific information listed above.
  2. Performance- light client side while server side has more of a heavy load.
- System
  1. Hardware
  2. Software- Python, JavaScript, and some various libraries
  3. Database- MySQL

- Security

  We will have encryption for the data being sent from the model to the website.

3 Project Specifications

- Focus / Domain / Area

  Potential home buyers

- Libraries / Frameworks / Development Environment

  JavaScript Libraries, Numpy, Pytorch, React.js, Visual Studio.

- Platform (Mobile, Desktop, Gaming, Etc)

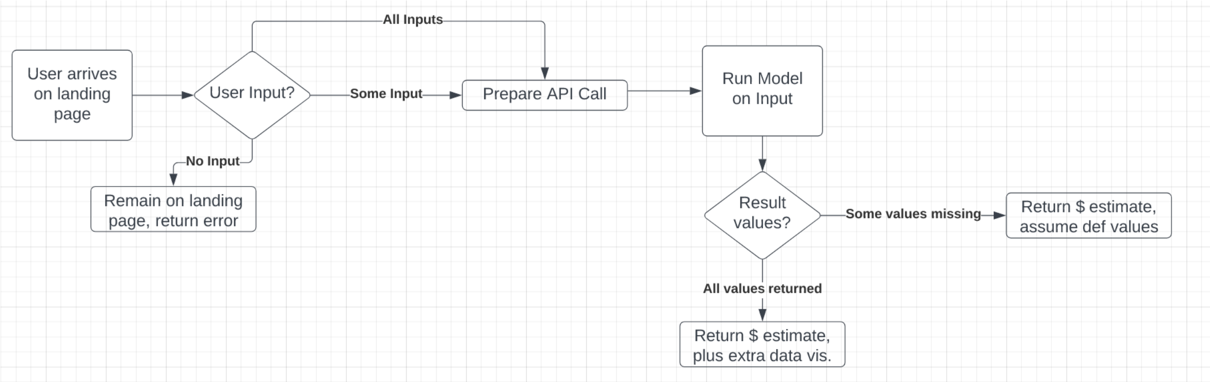
  Both Desktop and Mobile

- Genre

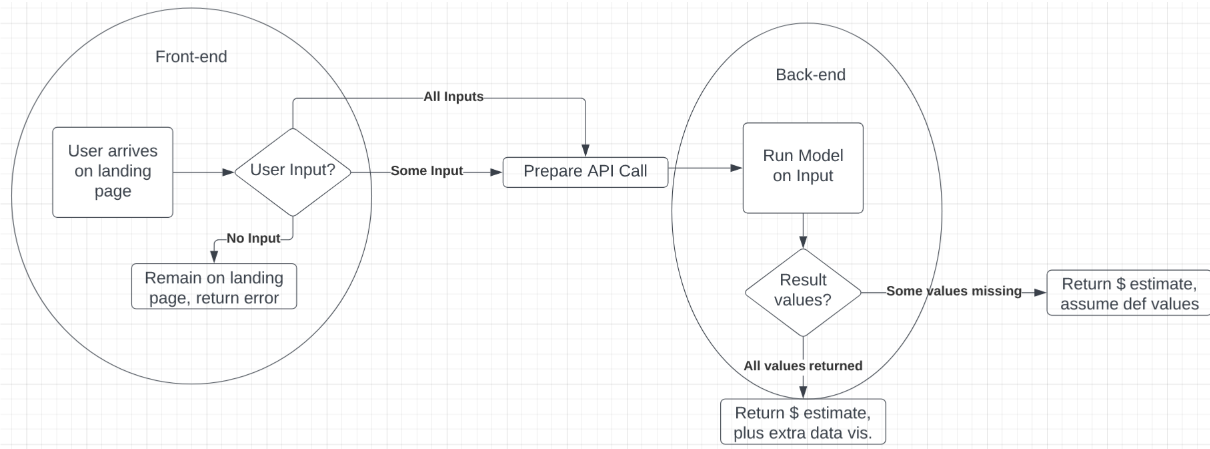
  Application

4 System Design

- Identify subsystems – design point of view
- UML Diagram(s)

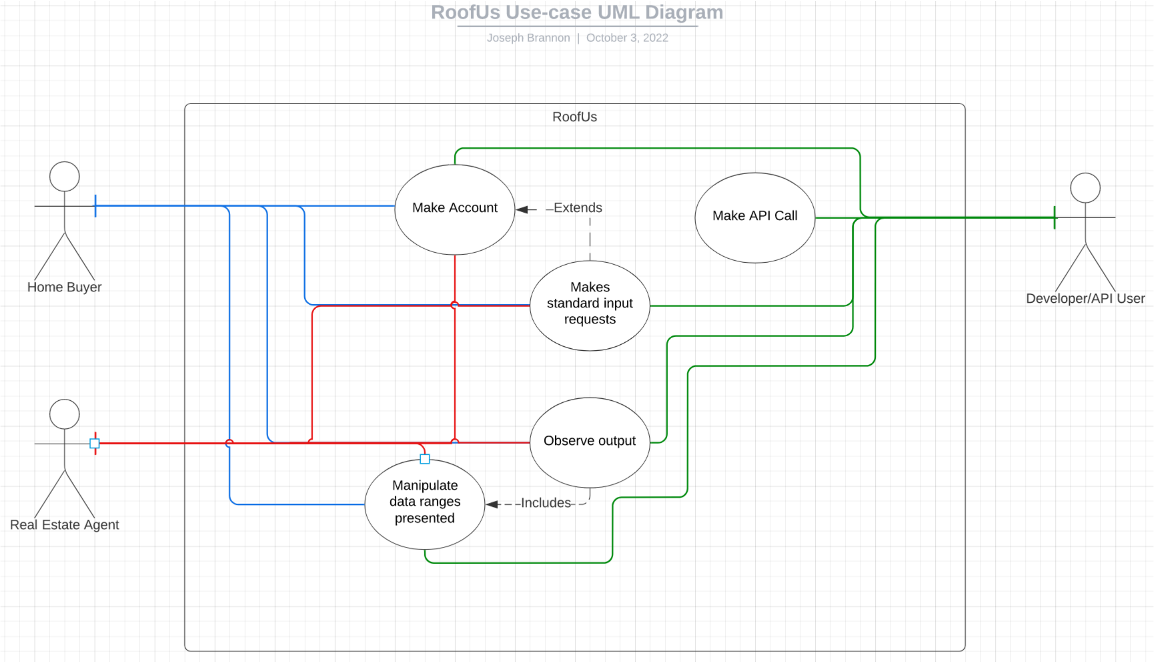

All Inputs
User arrives on landing page
User Input?
Some Input
Prepare API Call
Run Model on Input
No Input
Remain on landing page, return error
Result values?
Some values missing
Return $ estimate, assume def values
All values returned
Return $ estimate, plus extra data vis.

Description: Our chart is read flowing left to right. A user arrives on the site's landing page, and the site responds logically flowing based on the inputs provided from the user. For a simple base case: Should we come across no apparent input the user will receive an error and remain on their current page. If we do receive some type of input from the user we put whatever information we can into an API call that we can send from our client to our backend, and parse whatever data is available. In an ideal situation a user would provide all values and we wouldn't need to consider partial situations, but situations arise where a user may only know or be curious about one variable or input in particular such as Square Feet or Price. In such a scenario whatever information is not provided or parsable we will assume default values to answer a user's request to the best of our ability.



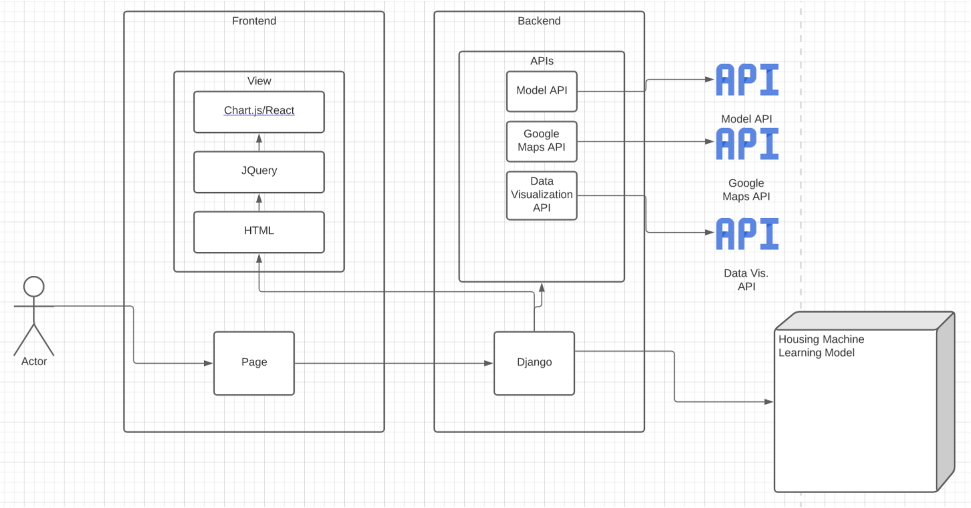
Description: This chart is a direct translation of the previous chart, but consideration is given for individual actions relative to front/back-end decision making. This is particularly useful as it gives a further illustration of both the theoretical running process of the website and helps to divide the labor and expectations of both the user/client and the backend/server.

- Use-case Diagram


RoofUs Use-case UML Diagram
Joseph Brannon | October 3, 2022
RoofUs
Home Buyer
Real Estate Agent
Developer/API User
Make Account
Extends
Make API Call
Makes standard input requests
Observe output
Manipulate data ranges presented
Includes

Description: This chart consists of 3 primary groups of end users for RoofUs. Color coding has been used to help clarity. For our 3 groups we have Real Estate Agents (Red), Developers/API end users (Green), and Potential Home Buyers (Blue). All 3 groups have access to the 4 key functionalities of the website: Make standard inputs, observe the system's response in order to display meaningful data in ranges, alter or modify the ranges used to view the data, and create accounts. These are all examples of universal features all sets of users may expect from the site and all potential user groups should have these controls by the nature of our Minimum Viable Product and Site Specifications (See Project Specifications on Page 3).The last and final use-case is only available to developers and API users which is to make calls to our API and get responses back via REST API. We believe offering a public facing API will both improve the longevity of RoofUs and increase the appreciability and potential audience of the site.

- Entity Relationship Model (E-R Model)
- Overall operation - System Model



5 System Analysis

- Data Dictionary

Data Source	Variable	Data Type	Description
DataVisualizationExample	dataFrame	dataframe	General datafra
contains demonstrations of data visualization	onlyNCHouses	dataframe	View of dataFra
	onlyGboroHouses	dataframe	View that only s
	onlyZIPHouses	dataframe	View that only s
	datalabels_arguments	dict	List of relevant
	state_options	dict	List of options r
	city_options	dict	List of options r
	zip_options	dict	List of options r
	stateChart	chart	Chart made fro
	cityChart	chart	Chart made fro
	zipChart	chart	Chart made fro
RoofUsHousingModel	dataFrame	dataframe	General datafra
primary backend model and variables	df	dataframe	Copy of dataFra
	category_features	list	list of strings co
	labels	Series	Series containin
	training	dataframe	dataframe of ou
	dfView	dataframe	Custom view of
	indexSeries	Series	Series containin
	chart	chart	Chart meant to
	X_train	dataframe	Portion of datas
	X_test	dataframe	Portion of datas
	Y_train	Series	Portion of datas
	Y_test	Series	Portion of datas
	train_dataset	dataframe	view containing
	test_dataset	dataframe	view containing
	every_column_except_y	list	list of columns u
	train_X	dataframe	dummy dataset
	test_X	dataframe	dummy dataset
	train_Y	Series	Series containin
	test_Y	Series	Series containin
	non_categorical_columns	list	List of columns
	numeric_features	dataframe	view of dataFra
	model	XGBRegressor	Our ML model
	most_relevant_features	list	list containing f
	y_pred	list	list containing t
	predictions	list	list of rounded i
	accuracy	float	float containing
	errlist	list	list of abs value