

Software Engineering of Web Applications

Final Project - Preliminary Presentation

Instructor: **Dr. Shiyu Zhou**

Group 8

Anjanibhargavi Ragothaman

Antons Treikalis

Anurag Ekkati

Priya Padmanaban

Sneha Bojja

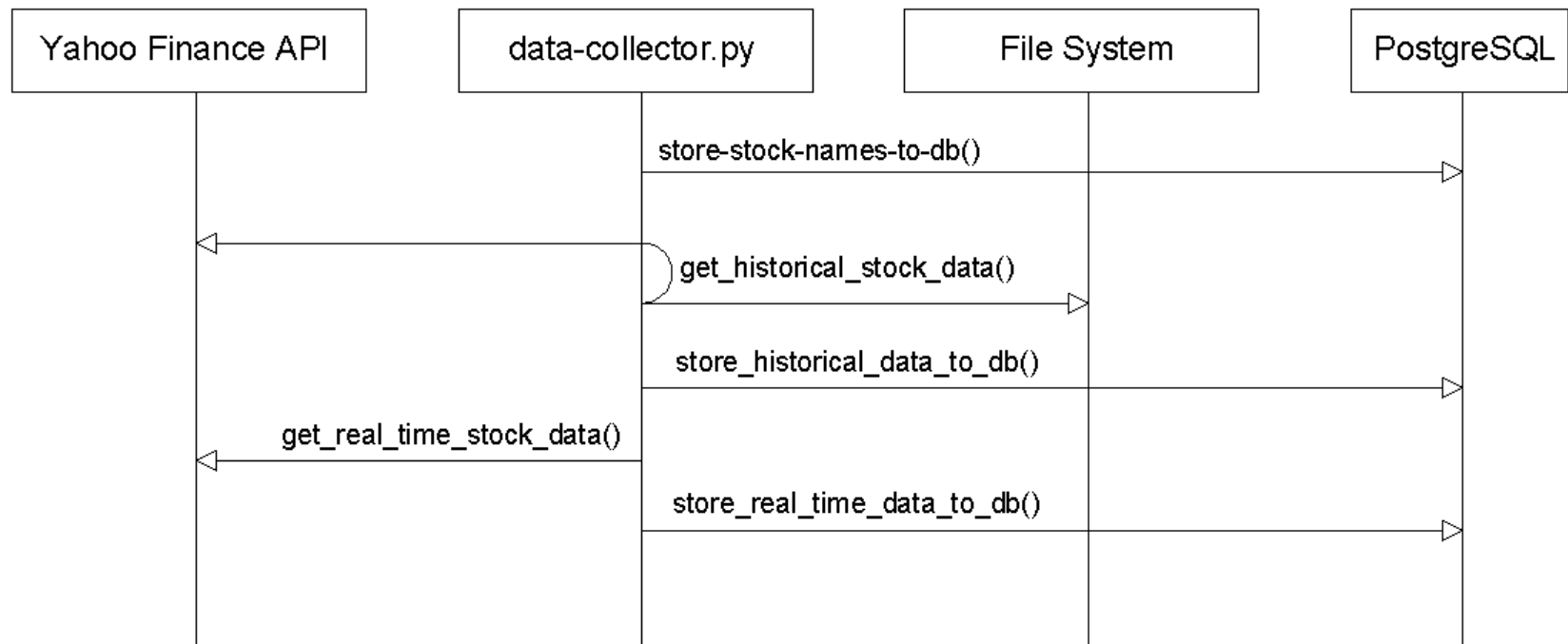
Background – Existing Systems

- Metastock
 - Over 250 built-in trading indicators and systems
 - 42 Adaptive indicators which use dynamic parameters
 - Templates that simplify applying the same set of indicators and studies to different securities
 - 9 Core Chart Styles
 - Bars, Line, Candlesticks, Point & Figure, Kagi, Renko, etc.
- Ninja Trader
 - Advanced Trade Management (ATM):
 - Automated stop loss and profit target order submission
 - Automatic self-tightening trailing stops
 - Automatic breakeven stop loss
 - Trade execution directly in a chart
 - Process Trading Signals from External Applications
 - Route orders from TradeStation or Esignal
 - Multiple interface options such as File, DLL and .NET

Achieved Tasks: Phase 1, Data Collection Module

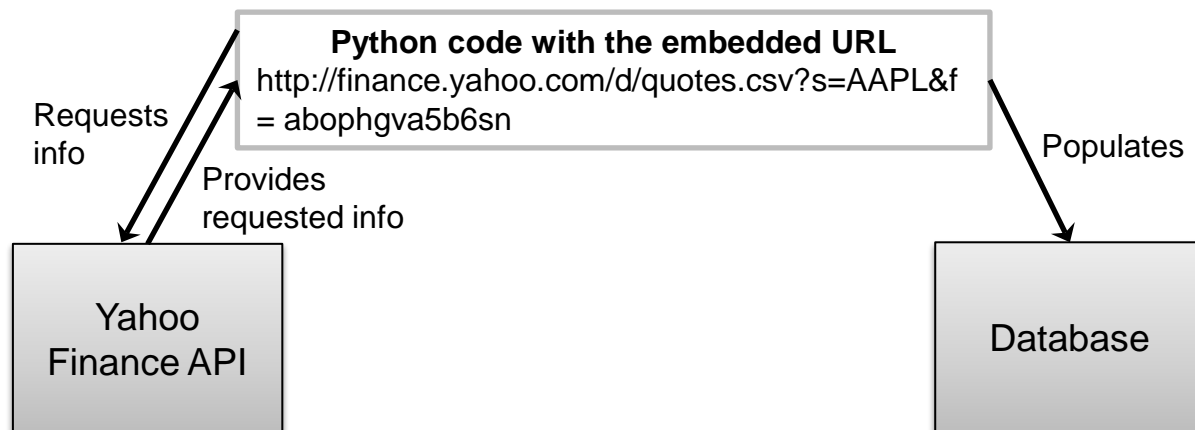
- Functionality to query Yahoo! Finance API
- Historical data collection for 10 stocks (Google Inc., NetApp Inc., etc.)
- Real time data collection every 60 seconds (for 10 stocks)
- Collected data stored in database – PostgreSQL
- Persistent environment for data collection – OpenStack VM
- Functionality to store historical / real-time data on local file system in csv format

Achieved Tasks cont. - UML control Flow diagram



Web sources – Yahoo! Finance API

- **REST** based API
- Great and simple way to download stock information
- Provides both **real-time** and **historical** data
- Returns stocks information in **csv format**
- Python program dynamically populates database with stock values from Yahoo Finance API
- Example

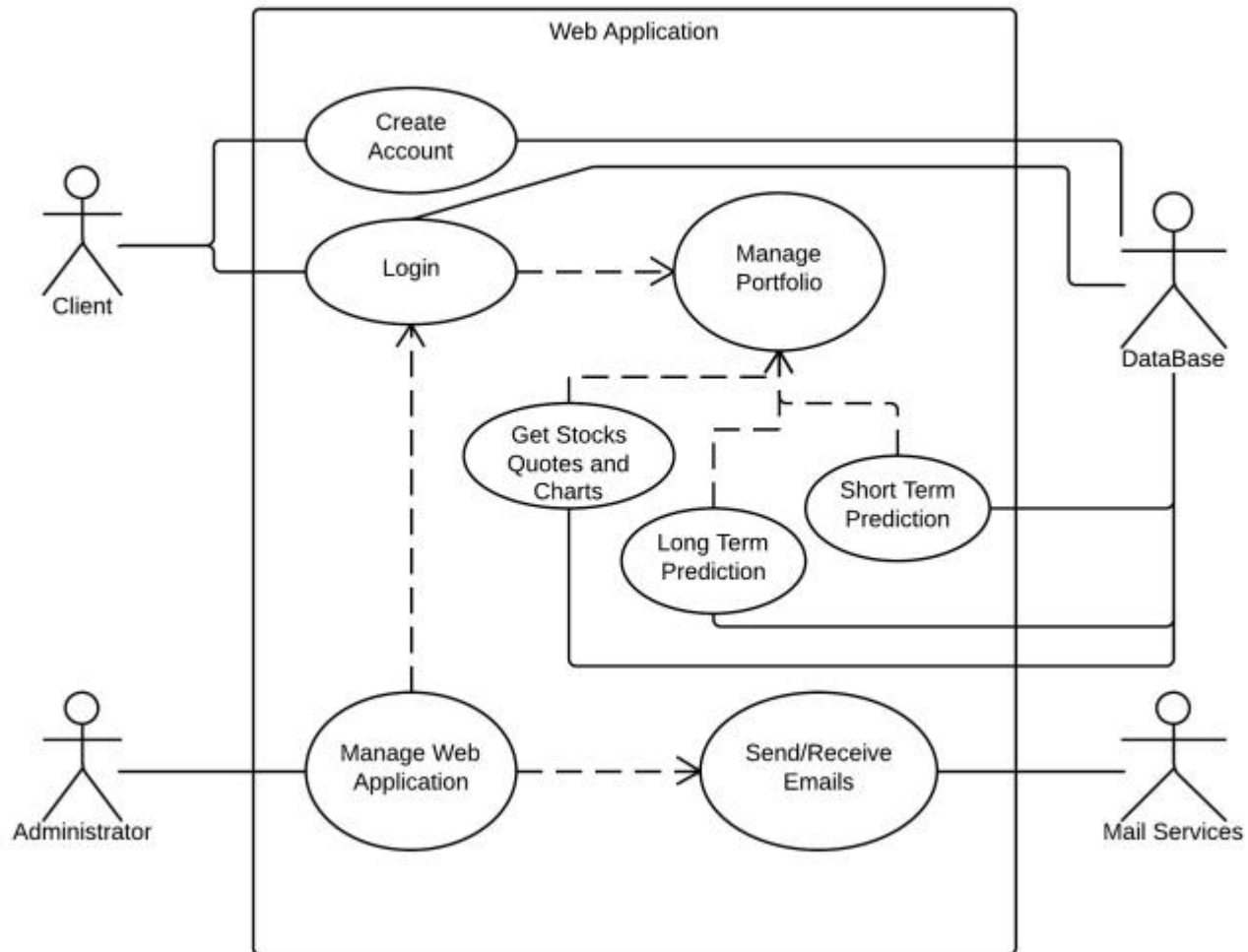


Distinctive features of this Project

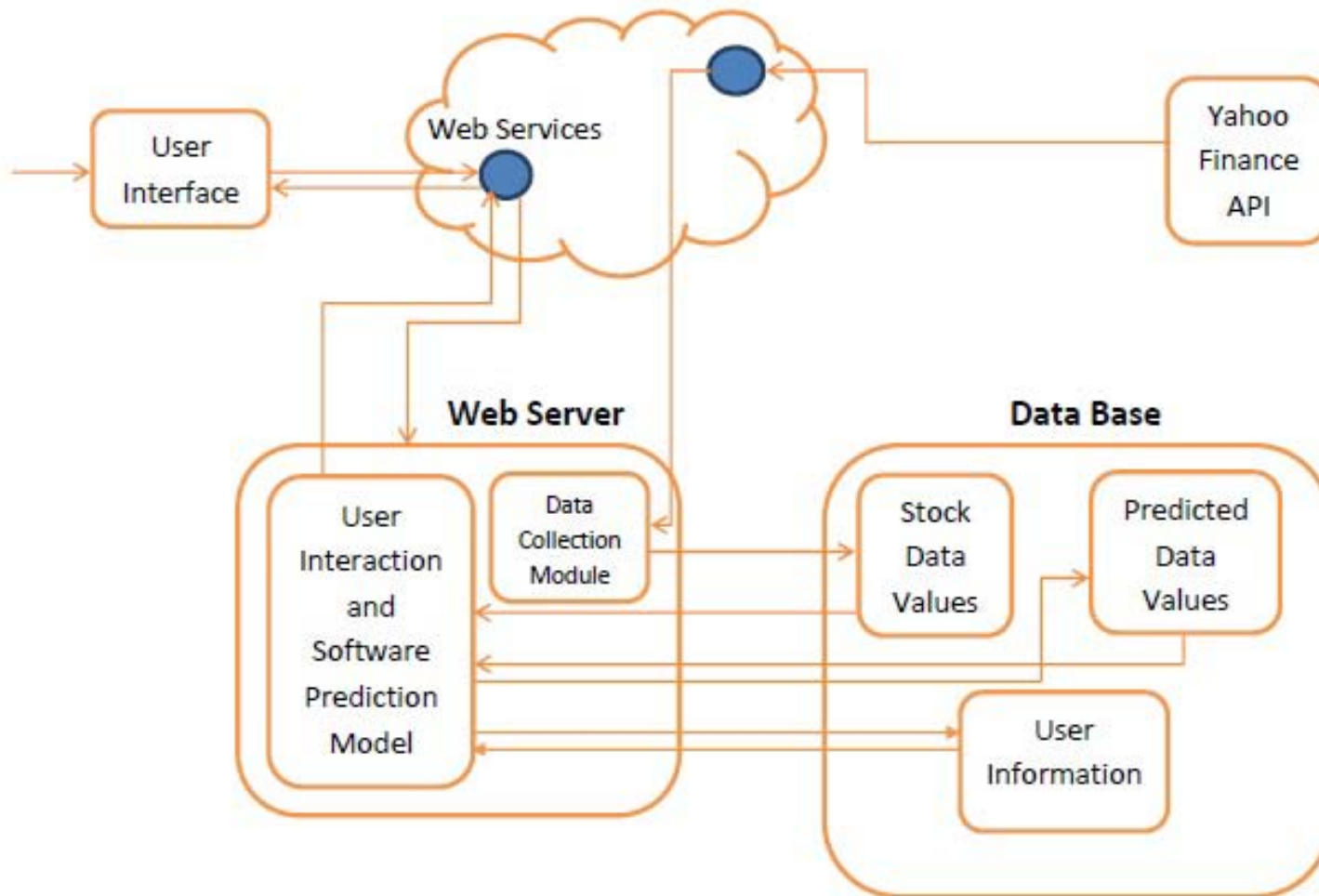
- Software Environment
 - Python
 - Django - Web Development Framework
 - Python Scikit - Machine Learning Library
 - PostgreSQL database
- Hosting
 - OpenStack VM on Future Grid
- Long term stock prediction with:
 - Support Vector Machine (SVM)
 - Gradient Boosted Regression Trees (GBRT)
- Stock price comparison with computed Moving Average to provide Buy/Sell suggestion to users*

* If time permits, comparison with Dow Jones Index will be included

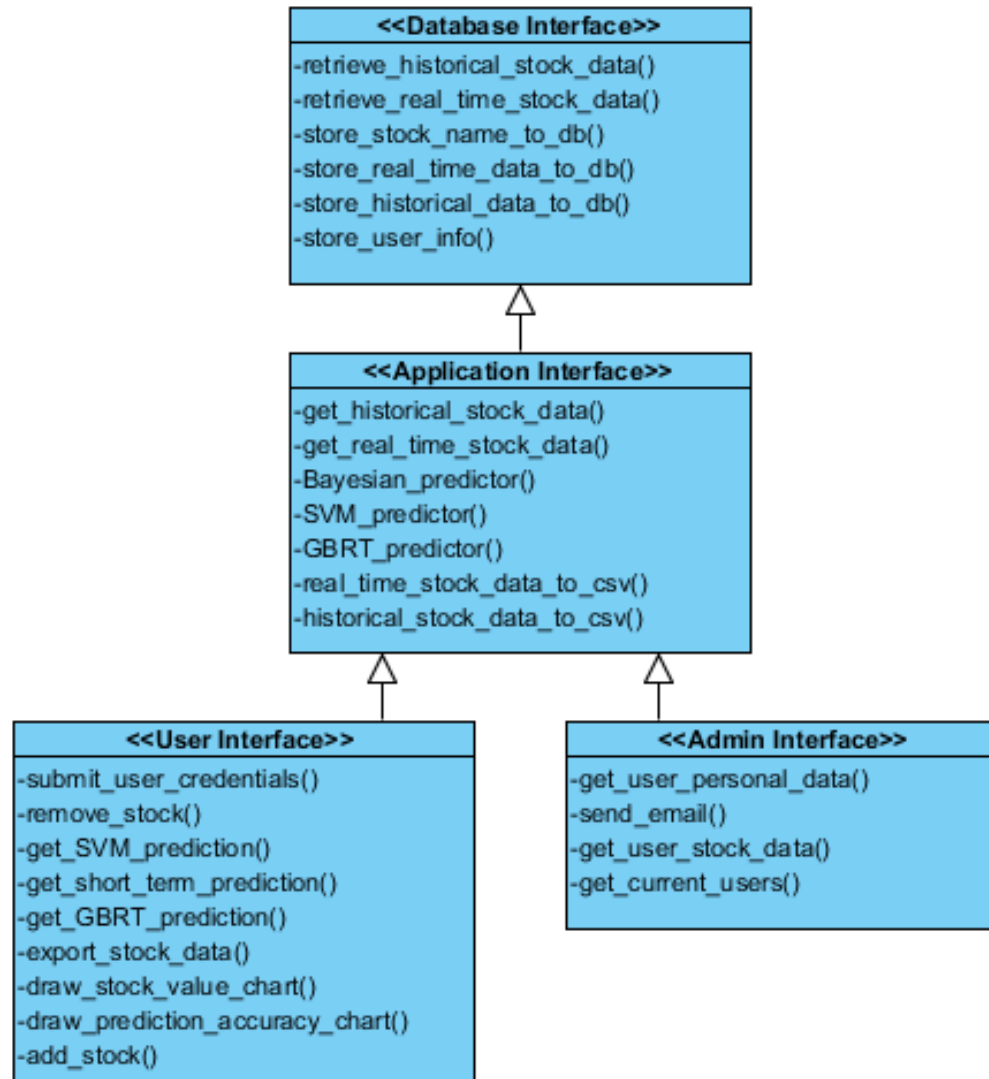
Use Case Diagram



Block Diagram



Web service interface



Short term Prediction Strategy - Bayesian

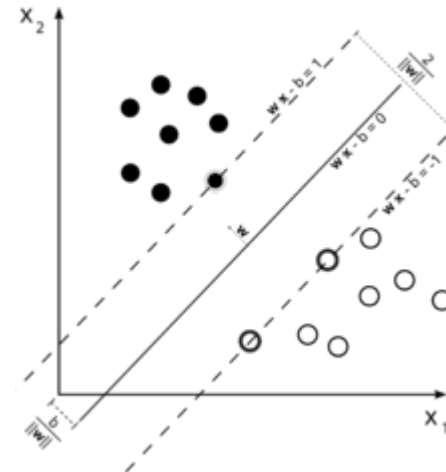
- Essential for Day-Trading purposes
- Predict Intra-day changes in stock prices
- Major Indicators Used:
 - Ask Price
 - Volume
- Other Indicators:
 - Moving Average Convergence Divergence (MACD)
 - Average for every 5 minutes
 - Force Index Indicator
 - $FI = [\text{Close (current period)} - \text{Close (prior period)}] \times \text{Volume}$
- Duration of prediction
 - 15 minutes
 - 5 minutes
 - 1 minute

Long Term Prediction Strategies

- Provides good starting point for stock market analysis
- Major Indicators Used (Used as input to predictors):
 - Daily Closing price
 - Volume (Index of Consumer Confidence Index)
- Other Indicators Used:
 - 50 days Moving Average Convergence Divergence (MACD)
 - 10 weeks MACD
 - Force Index Indicator
 - Helps to suggest Sell/Buy options to User
 - Predicted value Above MACD (Good to sell) / Below MACD (Good to buy)
- Duration of Prediction:
 - Monthly (Uses 20 years of price history when available)
 - Weekly (Uses 5 years of price history)
 - Daily (Uses 12 months of price history)

Long Term Prediction Strategy 1

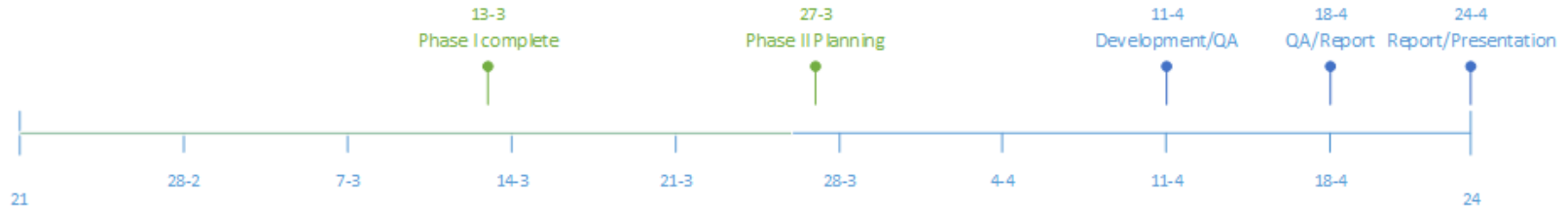
- Support Vector Machine - SVM:
 - Very resistant to over-fitting problems
 - Achieves High Generalization performance
 - Avoids the danger of getting stuck at local minima(non-linearity issue)
- SVM in simple terms:
 - Given a training data set, classify the data to one of the two categories
 - Good separation is achieved by having larger margin to hyperplane and training data
- Major Indicator Used
 - Closing price



Long Term Prediction Strategy 2

- Gradient Boosted Regression Trees (GBRT)
- ML technique for regression problems
- Prediction model in form of decision trees
- Generalization of boosting to arbitrary differentiable loss functions
- Pros:
 - Natural handling of data of mixed type (= heterogeneous features)
 - Robustness to outliers in input space (via robust loss functions)
 - Requires less training data than SVM
- Cons:
 - due to the sequential nature of boosting is difficult to parallelize
- Implementations:
 - TreeNet (TM Salford Systems, Inc.)
 - MART (TM Jerill, Inc.)

Work plan - Gantt chart



Index page and User Home page

User Profile Page

Short term and long term prediction pages/fields

Moving Average prediction for daily/weekly fields

Writing database API and improving data collection module

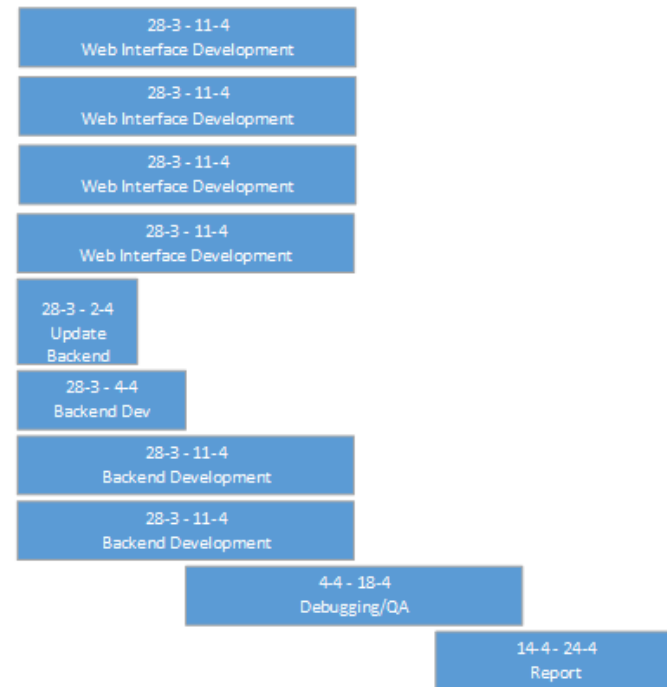
Integration of Bayesian predictor for short term prediction

SVM Prediction code development for long term prediction

MART Prediction code development for long term prediction

Testing and Debugging of Backend and Frontend modules

Report Writing and Final Presentation



Contribution Breakdown

Task	Ragothaman	Treikalis	Ekkati	Padmanaban	Bojja
Front end Web interface:					
Index page and User home page			X		
User profile page				X	
Short term and long term prediction page/fields				X	X
Moving Average prediction for daily/weekly fields					X
Comparison with Dow Jones Index field			X		
Testing and Debugging Web Interface	X	X	X	X	X
Back end development:					
Writing database API / improving db			X		X
Integration of Bayesian predictor for short term		X			
SVM Prediction code development for long term	X				
GBRT Prediction code development for long term		X			
Testing and Debugging back end module	X	X	X	X	X
Report writing:	X	X	X	X	X
Total percentages:	20%	20%	20%	20%	20%

Thank you for your attention!