

Group 3: Auto Park

Contributions Breakdown:

We do not have integration testing in our project because we are just planning on having a website that both customers and management can access from any browser. The points from this section should be redistributed to the code section.

In addition, Chunhua Deng, Siyu Liao and Xianglong Feng proposes an offline data analysis and price simulation process which interacts with the website and garage system via hourly rate setting. Moreover, they have also reproduced the model from a published work (Tian, Qiong, et al., 2018) which is found purely theoretical and thus they propose their own model given the real world data.

Equal Contribution Sections:

- Demo Slides
- Reports

Chunhua Deng

- Research
 - Search the state-of-the-art dynamic parking systems
 - Reproduce results from published paper (Tian, Qiong, et al., 2018)
 - Real World parking demand modeling via regression
 - Calculate the price based demand function with multiple model.
 - Calculate the time based demand function by regression on the real data.
 - Data fitting algorithm
 - Profit optimization based on demand function (30%)
 - Curve fitting algorithm research
- Garage module coding
 - Email notification module coding
 - ExitGate module coding
 - FlatRate module coding
- Smart Pricing Module Coding
 - Large data volume querying via spark (~62M records, ~5.3GB)
 - Hourly parking demand analysis
 - Daily parking demand analysis
 - Weekly parking demand analysis
 - Monthly parking demand analysis
 - Length of stay analysis
 - Plate Image Cropping
 - Price Page View in the reservation website
 - Billing Function in the garage system

Corey Chen

Requirements Specifications

- Register (UC-1) implementation
- Update Information (UC-11) implementation
- Reservation and Cancel Reservation implementations (UC-3 and UC-7)

Software Design:

- Registration Module (logic and database queries: 60%)
- Routing between pages on the site (Debugging routing, logic, and formatting issues)
- Reservation and Cancellation Pages (50%)
- Update user information Page (100%)
- Front end (formatting the website pages: 50%)

Coding and Logic:

- Create/Cancel Reservation Modules (40%)
- Update User information (100%)
- Registration Module (60%)
- Login Module (Debugging and Testing: 40%)
- Routing Testing and Debugging (50%)

Debugging/Testing:

- Unit Testing for Website (All pages: 50%)
- Website routing problems and Database interactions/ queries (40%)
- Database Management (40%)
- Database Table Organization and Configuration (40%)

Other:

- Organizing and scheduling meetings/ other logistics

Jonathan Garner

- SMS Notification Module
- Authentication Code Module
- Brochure (50%)
- Garage Logic (Debugging/Testing)
- Spot status reporting
- Parking map display
- Database communication
- Some website debugging
- Parking Map GUI

Ridhima Sakhuja

- Registration Module (logic and input validation) (40%)

- Interaction between login/registration pages
- Login Module (20%)
- Brochure(50%)
- Front End Formatting Website (50%)
- Account Management Module (front end) (50%)
- Cancel Reservation Page (logic) (50%)
- Website Routing (40%)
- Unit Testing

Siddharth Musale

- Entrance Gate Module
- Elevator Module
- Spot Verification Module
- Traffic Management Module
- Login Module (80%)
- All html view pages
- Page routing on web application
- Login and Reservation Logic
- Exit Gate, Billing, and Notification Module Integration
- Database Creation, Configuration, and Management

Siyu Liao

- Requirements Specification:
 - All Smart Pricing Related Requirements, mostly in section 4.4
 - REQ-2, REQ-15, (plate image cropping)
 - REQ-45 (dynamic price web page)
- Software design
 - Billing Function in the Garage System
 - Price Page View in the reservation website
- Coding
 - Large data volume querying via spark (~62M records, ~5.3GB)
 - Hourly parking demand analysis
 - Daily parking demand analysis
 - Weekly parking demand analysis
 - Monthly parking demand analysis
 - Length of stay analysis
 - Plate Image Cropping
 - Price Page View in the reservation website
 - Billing Function in the garage system
- Debugging/Testing:
 - Test the large data volume reading in the correct format
 - Test the spark hourly, daily, weekly, monthly RDD generated correctly

- Test the data filtering resulting in correct data
- Test the dynamic price function and revenue function is as in the paper
- Test the logic regression for raw data in daily, weekly and monthly.
- Tuning the parameters (the degree, the coefficient and so on) to best fit the data and reduce the error
- Test ease out quart function for price-based demand function in peak time
- Test ease out quart function for price-based demand function in adjustable time
- Test the solution to calculate the optimized price value in peak time
- Test the solution to calculate the optimized price value in adjustable time
- Test the derivation and optimization solution in calculating the benefits
- Research
 - Literature Review (25%)
 - Reproduce a model from published paper (Tian, Qiong, et al., 2018)
 - Motivation identification (our model has more practical meanings than published)
 - Real World Parking demand analysis
 - Real World parking demand modeling via regression
 - Exponential reaction assumption toward the dynamic pricing
 - Calculate the price based demand function with multiple model
 - Objective function definition/expression
 - Profit optimization based on the demand function. (30%)
 - Propose the solution to the demand function optimization in single module.
 - Proposed and design the dynamic updating method for price-based demand function
 - Propose and design the ease out quart function for price-based demand function in peak time
 - Propose the design the ease out quart function for price-based demand function in adjustable time
 - Design the solution to solve the optimized price based on time
 - Design the method to calculate the new demand function after modified by the price-based demand function

Xianglong Feng

- **Requirements Specification:**
 - All Smart Pricing Related Requirements, mostly in section 4.4
 - REQ-45 (dynamic price web page)
- **Software design**
 - Billing policy in the Garage System
 - Dynamic price algorithm
- **Coding**
 - Large data volume preprocessing (~62M records, ~5.3GB)
 - Hourly parking demand modeling for curve fitting
 - Weekly parking demand modeling for curve fitting

- o Monthly parking demand modeling for curve fitting
- o Billing policy in the garage system
- o ease out quart function for price-based demand function in peak time
- o ease out quart function for price-based demand function in adjustable time
- o The solution to calculate the optimized price value in peak time
- o The solution to calculate the optimized price value in adjustable time
- o The derivation and optimization solution in calculating the benefits
- o Final data visualization
- **Research**
 - o Literature Review (25%)
 - o Model Analysis, testing and validation from published paper (Tian, Qiong, et al., 2018)
 - o Motivation identification (clearly summarize the two goals: The best benefits & convenient for user)
 - o Analyze the drawback of previous work and analyze the potential improvement
 - o Real World Parking demand analysis and abstraction for demand function
 - o Propose multi-module with multi-model method for price-based demand function (the key idea for dynamic pricing algorithm.)
 - o Propose the solution to the demand function optimization in single module.
 - o Proposed and design the dynamic updating method for price-based demand function
 - o Propose and design the ease out quart function for price-based demand function in peak time
 - o Propose the design the ease out quart function for price-based demand function in adjustable time
 - o Design the solution to solve the optimized price based on time
 - o Design the method to calculate the new demand function after modified by the price-based demand function
 - o Profit optimization based on the demand function. (40%)
- **Testing:**
 - o Test the logic regression for raw data in daily, weekly and monthly.
 - o Tuning the parameters (the degree, the coefficient and so on) to best fit the data and reduce the error
 - o Test ease out quart function for price-based demand function in peak time
 - o Test ease out quart function for price-based demand function in adjustable time
 - o Test the solution to calculate the optimized price value in peak time
 - o Test the solution to calculate the optimized price value in adjustable time
 - o Test the derivation and optimization solution in calculating the benefits