Local Life Platform

Author: Zhonghua Bao, Anirudh Bezzam, Vignesh Mohan, Vineet Singh,
Shrey Vaity

Issued Date: Feb 05, 2020

INTRODUCTION

Local life platform is a proposed solution for both customers and business owners who come across bottlenecks with current products on the market. We admit killer apps among a sea of ecommerce platforms, like Amazon and GrubHub, received great success already and are continuously making difference to people's daily life, however, they are not perfect in every shopping scenario. To be specific, a customer needs to wait a relatively long time before his online orders are delivered on Amazon, and he is not able to buy items other than food from restaurants on GrubHub either. For the business owner, it is probably a problem to develop their own online service, or not eligible to register an online shop on a big platform, which is greatly hindering them to make a prosperous business. Therefore, a new product is considerable to shoot mentioned pain points, and that's the reason why we are working on this project.

With this platform, users are entitled to access to the following services: first of all, they can login the system, and different interfaces will be provided to customers and business owners separately. Business owners can register their shop and simultaneously manage their inventory. On the other hand, customers can search their desirable items, read and write comments, place an order. In order to make our platform seem to follow the trend, we want a feature of recommendation either, rendering items that the customer will be interested before he starts looking for something.

PS: The related requirement document, architecture document will be included in the next version.

ROLES AND RESPONSIBILITIES

In this project, we have following roles and their associated responsible people:

Development Lead:	Zhonghua Bao
Buildmeister:	Shrey Vaity
Architect:	Zhonghua Bao
Developers:	Zhonghua Bao, Anirudh Bezzam, Vignesh Mohan, Vineet Singh, Shrey Vaity
Test Lead:	Vignesh Mohan
Testers:	Zhonghua Bao, Anirudh Bezzam, Vignesh Mohan, Vineet Singh, Shrey Vaity
Documentation:	Zhonghua Bao, Anirudh Bezzam, Vignesh Mohan, Vineet Singh, Shrey Vaity
Documentation Editor:	Anirudh Bezzam
Designer:	Anirudh Bezzam
User advocate:	Vineet Singh
Risk Management:	Vineet Singh
System Administrator:	Shrey Vaity
Modification Request Board:	Vignesh Mohan
Requirements Resource:	Vineet Singh
Customer Representative:	Vineet Singh
Responsible for Acceptance Testing:	Vignesh Mohan
Machine Learning Expert:	Anirudh Bezzam

METHOD

· SOFTWARE

- **Front end:** HTML5, CSS3, JavaScript 8, React16.8, Bootstrap4
- **Back end:** Python3.7, Django 3, MongoDB
- **Operating System:** Windows 10
- **Software Packages:**
- **Code conventions:**
    - **Front end:** Airbnb JavaScript Style Guide
    - **Back end:** Django Coding Style

· HARDWARE

- **Development Hardware:**
- **Test Hardware:**
- **Deployment Hardware:** Amazon Web Service EC2

· BACK UP PLAN (INDIVIDUAL AND PROJECT)

Everyone is asked to fetch the latest version of the project, and start his own work locally, finally make a pull request when his work is done. So we can say each member has his own back-up to his role on their respective local machine.

As to the project, the code is saved on GitHub, so that we can clone at any time, and we can also select some components from the previous version to make a new build.

· REVIEW PROCESS

We will absolutely conduct the architecture and design review, because it matters a good implementation of our system, especially when the previous technical solution will not be inherited. Overall discussions will be continuously arranged until a satisfied entire image has been approved. And Even if one is not responsible for a certain feature, he still needs to work together for a module's work, and how they should be organized.

Since we are newbies to the technology stack we picked up, we think it would be quite challenging if we still want code review. We try to guarantee a good level in usability, and security via AWS kits.

Informal approaches are favorable for this project, because a workable software is more important this time rather than a high standard industrial product. Formal negotiation is prone to limit our time is learning frameworks and making contributions.

Among our team, Zhonghua Bao will be responsible for reviews and resolving any issues unveiled by reviews. More importantly, every team member should bear in mind the coding style conventions very well and try their best to come up with codes in good quality.

· BUILD PLAN

- Our code are saved on a Github repository
- Every member pushes their new code immediately once they are done, and a continuous integration will be triggered by Travis CI pipeline.
- When a feature is ready, test the functionality and the usability really quick, and then make sure other feature is still working well in the testing environment
- Deploy the feature on AWS cloud service.

· MODIFICATION REQUEST PROCESS

- **Modification Request tool:** Modification requests are submitted as a GitHub Issue tracker.
- **Decision process:** Everyone's approval is required in this case, because it's quite compact to finish the development plan. If so, this request will be split into several user stories, and placed them in the sprint planning list. We will update the corresponding part in documents either.

VIRTUAL AND REAL WORK SPACE

We have a mix of both virtual and real workspace. We meet at least once physically on Thursday, so that we can know everyone's progress, and shoot some problems together. Virtually, we have a short meeting on the weekend on Google meetup and create a slack team for enhanced communication. This worked quite well last year. We can set some goals for the coming week, and solve some problems either.

COMMUNICATION PLAN

"HEARTBEAT" MEETINGS

The most critical meeting weekly is a video conference on Google meetup, and everyone's attendance is mandatory. It is scheduled on the weekend, because we think it can help us have a good prep for the coming week, and we are not rushed to get something done immediately. During this meeting, we are going to talk about the completeness of the goal set last week, then we can be clear what blocks us to move forward and try to solve it asap. Finally, we need to declare the development plan next week.

STATUS MEETINGS

Status meeting will be scheduled on Thursday before or after we talk with Professor Kempinski. Since it's the middle point of a week, and we can expect a rough weekly outcome we will yield. If the status is not very satisfied, a much harder effort should be exerted to make sure we follow on the progress. And also, if we already come across some obstacles, it can be a good time to clear.

ISSUES MEETINGS

Issue meetings can be scheduled at any time when people are available, as long as it can help solving problems, and not everyone is required to join.

TIMELINE AND MILESTONES

Below is the list of milestones for this project:

February 5th	Project DevPlan
February 12th	Architecture and Design
March 5th	User Login
March 19th	Database, Business Registration, Inventory Management
April 2nd	Order Placement, Comments
April 16th	User Interface, Recommendations
April 30th	Final Demo

TESTING POLICY/PLAN

We are highly freaky with unit testing, as this ensures the functionality from the least module. Any serious mistake can be prevented from very trivial practice. Everyone must test their code carefully before they make a pull request and committing new code to the master branch.

As long as there is a new push, an automatic build will be triggered by Travis CI, this will pass the unit tests listed on the script. After that, we mainly focus on the API test, as we want to make sure every feature is functioning well after some modules are put together. This will be carried out by Postman. Also, we will test any existing feature after a new feature has been incorporated.

RISKS

- People cannot learn the new knowledge quickly
- We don’t have an outstanding carrier to ensure the lower limit
- A good dataset takes a long time to fabricate or find

ASSUMPTIONS

We assume every member just has basic knowledge in web development, and needs to take some time before they write the real first line of code to this project. And, everyone will be responsible for one feature or two, instead of someone who is in charge of the front end, or anyone else gets the backend done.

DISTRIBUTION LIST

1. Professor Rich Kempinski