

Attendees

- Anandan, Taksh, Sonic & Vidish Mehta

Thank you all for attending, we had a productive session discussing all the important and actionable details for our project ☺

Discussion Items

- Front-end System Update
 - UI form for users to submit their request. The UI form will contain three important parameters:
 - Pre-defined proximity parameters based on user preferences such as school, workplace, supermarket, coffee shop, MRT accessibility, gyms etc.
 - Prototype development with pre-loaded parameter location list based on tested district (for example, if our prototype development focuses on the east, for schools, we can put in UWC international school, Bedok Primary/Secondary School etc)
 - User-defined weights for user to provide proximity preference of certain locations (priority 1 → highest, 9 → lowest). User is allowed to provide the same proximity parameters for his preferences in which case algorithm will decide purely based on difference.
 - For beautification (least priority now), google map to be added below the form which provides real-time location of each of the user selection. (Example of the front-end UI is available at the end of the notes)
- Backend Algorithm Updates
 - Backend algorithm consists of a scoring mechanism to score each of the current node, search space definition and distance calculation mechanism.
 - Scoring Mechanism – Least score is most promising node. For example, for starting node A (could be inverse as Taksh suggested both give the same results).

User-Defined Function	Preference	Distance from current node A
School	2	7 km
Restaurant	5	2 km
Supermarket	4	3 km
Workplace	3	4 km

Score: $(7*2)+(2*5)+(4*3)+(3*4) = 48$

- Confound Search Space – can be found by finding creating a square grid based on locations with largest and shortest latitudes and longitudes. Node in each of the grid can be arbitrarily defined by the backend developer (eg. a 100*100 square).
- Multiple search trees with different starting nodes to be evaluated using the scoring mechanism. Computational scalability to be provided by multi-processing (Taksh will be guiding us on how to achieve this). Time-dependent search with the best results to be provided to the user.

- Distant calculation mechanism → simple prototype testing, Euclidean distance. Actual road distance, mrt distance can be evaluated based on success of initial prototype.
- Tech Stack
 - Frontend systems
 - Technology
 - HTML, CSS (bootstrap) & Angular JS based on familiarity and leveragability of code
 - IDE
 - Visual Studio Code
 - Backend Systems
 - API
 - Google Maps API (check out googlemaps python libraries which gives latitude and longitude location)
 - Rental API for Singapore (check out squarefoot api, haven't tried it but it should be what we will need)
 - Algorithm development
 - Python
 - IDE
 - Visual Studio Code / Spyder

Action Items Update for next meeting (Monday → 3/16/2020)

- New VM for integrated project development → Taksh to help create a new VM.
- Github Repository Creation → Anandan
- Google API testing & Rental API testing → Anandan & Sonic
- Algorithm & Backend testing & development → Vidish & Taksh

Frontend UI Example (simplified)

Rent@Us

	Location	Weight
School	<input type="text"/>	<input type="text"/>
Workplace	<input type="text"/>	<input type="text"/>
Restaurant	<input type="text"/>	<input type="text"/>
Supermarket	<input type="text"/>	<input type="text"/>
Mrt Accessibility	<input type="text"/>	<input type="text"/>

