NPS2001C: Matrix Unplugged:Using Computer for Real-World Problems [2320]

GROUP PROJECT MILESTONE 2

Lecturer: Dr Jonathan Kang

Group Members:

A0253501U Ethan Lim Jun Xian

A0254490A Chan Junlin, Dylan

A0253153M Isaac Lim Shile

A0252179Y Ang Rui Jie, Thaddeus

FlushFinder: Data Report

This report highlights the key data FlushFinder requires to function and how said data will be processed. We will also discuss the data privacy policies and cybersecurity measures we will employ. FlushFinder will primarily use **user generated data** and **toilet information data** provided from our side.

FlushFinder will leverage an existing database of toilet locations obtained via the OneMap API. A separate database containing the respective toilets' facilities will be linked to accommodate the user's preferences during the search. FlushFinder then requires user generated data in the form of ratings, reviews, and preferences for the subsequent recommendations that will be presented to said user. This review data will then be sent to the Item Based Collaborative Filtering (IBCF) algorithm to recommend future toilets that the user may prefer.

After several usages, FlushFinder will present two lists for the user to choose upon their next search request: Toilets according to the search fields inputted by the user, toilets the user may prefer based on prior experiences generated from the IBCF algorithm. The following is a summary of the data fields, and how it will be used in the application:

User-Generated Data:

User Account Data

FlushFinder will utilise our user's preferences to find a toilet based on their needs and wants. Our users will create accounts using their email addresses and create a unique password, so that they can post toilet ratings after their usage. Having ratings tied to a unique account adds a layer of integrity to whether the reviews are made by genuine users. Users with no accounts can only search for toilets according to their preferences but cannot review.

User Preferences

During the search, FlushFinder will ask users to provide their preferences for toilet features by checking a box next to the feature(s) they want. Five features will be provided: *Bidet, Wheelchair-Friendly, Sanitary Pad Dispenser, Water Cooler, and Nursing Room*. This data will only be used to personalise toilet recommendations during the search phase.

User Location Data.

While using the app to find nearby toilets, the user's real-time location will be collected. The user will use a slider to choose how far they want the search to take, from a minimum radius of 200 metres and up to 1 kilometres.

User Ratings and Reviews

Users can optionally provide a rating and written review after using a toilet found through the app. This rating and review is based on three categories: *cleanliness, quality of amenities, and*

overall ambiance. We will ask the user to rate the toilet according to these categories on a scale of 1 to 5.

User Item Matrix (IBCF)

The user's item-matrix is a unique matrix required for the IBCF algorithm to function. This matrix includes the frequencies of amenities used by a specific user, collated from their rated and reviewed toilets. This ensures that the item-matrix is only updated by toilets the user actually used and reviewed, for more accurate recommendations via IBCF.

Toilet Information Data:

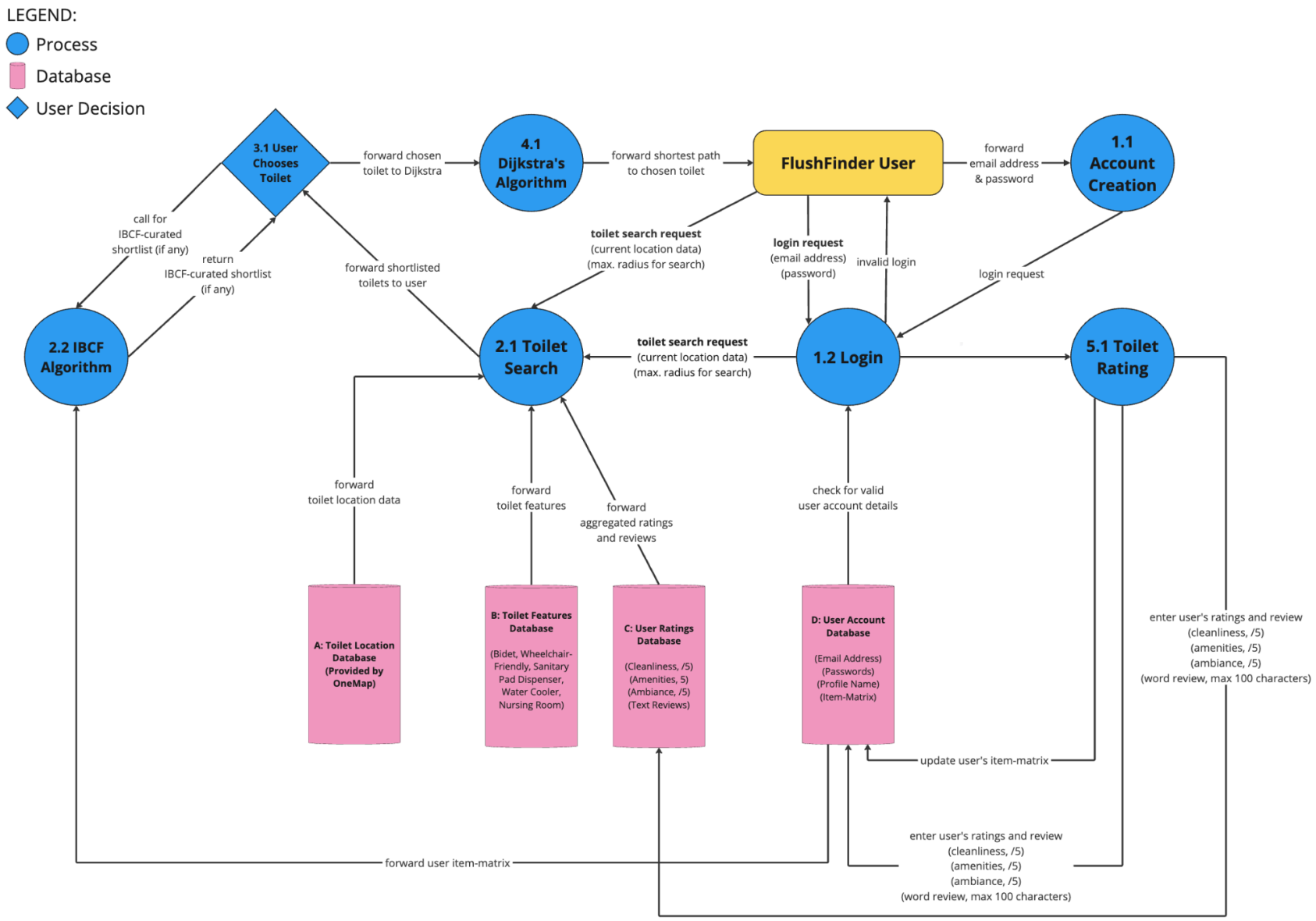
User Item Matrix (IBCF)

FlushFinder will leverage on the OneMap API provided by the Singapore Land Authority for open-source geolocation data on the publicly available toilets. The locations will be fed through Dijkstra's algorithm together with the user's current location to determine the shortest possible path to the chosen toilet.

Toilet Features

A database containing the availability of features listed in the 'user preferences' above per toilet provided by the OneMap API. This data will be called during the toilet search process to shortlist toilets for the user.

The following page showcases our data flow diagram.


LEGEND:
Process
Database
User Decision
3.1 User Chooses Toilet
forward chosen toilet to Dijkstra
4.1 Dijkstra's Algorithm
forward shortest path to chosen toilet
FlushFinder User
forward email address & password
1.1 Account Creation
call for IBCF-curated shortlist (if any)
return IBCF-curated shortlist (if any)
forward shortlisted toilets to user
toilet search request (current location data) (max. radius for search)
login request (email address) (password)
invalid login
login request
2.2 IBCF Algorithm
2.1 Toilet Search
toilet search request (current location data) (max. radius for search)
1.2 Login
5.1 Toilet Rating
forward toilet location data
forward toilet features
forward aggregated ratings and reviews
check for valid user account details
A: Toilet Location Database (Provided by OneMap)
B: Toilet Features Database
(Bidet, Wheelchair-Friendly, Sanitary Pad Dispenser, Water Cooler, Nursing Room)
C: User Ratings Database
(Cleanliness, /5)
(Amenities, 5)
(Ambiance, /5)
(Text Reviews)
D: User Account Database
(Email Address)
(Passwords)
(Profile Name)
(Item-Matrix)
enter user's ratings and review
(cleanliness, /5)
(amenities, /5)
(ambiance, /5)
(word review, max 100 characters)
update user's item-matrix
forward user item-matrix
enter user's ratings and review
(cleanliness, /5)
(amenities, /5)
(ambiance, /5)
(word review, max 100 characters)

FlushFinder: Security Policies

Data Sharing

No personal data, including user preferences and location history, will be shared with or sold to any third parties for secondary purposes like advertising or research. Anonymized, aggregated statistics and trends, such as average ratings by toilet or top search locations, may be shared with toilet providers or public planning agencies to help improve toilet access and quality. However, this will contain no user-level personal or re-identifiable information. A similar stance has been adopted with regard to our use of the IBCF algorithm, the data used to produce recommended toilets to users will be kept entirely anonymous.

By only collecting necessary data, using it solely for core app functions, and never sharing personal information externally, FlushFinder aims to deliver personalised recommendations while fully respecting user privacy. All data practices will be transparently communicated in the app's privacy policy.

Data Privacy

FlushFinder will need to collect and store some personal data such as user location and preferences to provide personalised toilet recommendations.

User Account Data

This data is kept by servers for verification purposes. First time users will be emailed directly an OTP (one time password) for activation of their account. This email data will be also used for future login verification. Passwords will be encrypted through password hashing to protect user information.

User Location Data.

User’s live location will need to be collected and used in order to find nearby toilets. Firstly, the users’ consent will be requested to allow for data collection before they are able to use the application. Users will be able to revoke their consent if they want to. Once consent is given, their coordinates data will only be used temporarily for locating toilets and providing navigation. It will not be stored long-term or linked to their user profile.

User Ratings and Reviews

Only the rating and review text will be stored, not the user's identity, this anonymised data will be made publicly available on the app for other users to see. This protects users anonymity whilst sharing their ratings to the larger FlushFinder community. Whereas privately stored data like amenity use would be tied to users in the user-item matrix, will be used in the IBCF recommendation algorithm. This would make users’ amenity choices processed by the algorithm but hidden from the public.

Example

Taking into account all the above, here is a generalised step-by-step example of how user data is handled:

1) Jonathan Kang creates an account on FlushFinder with the following details:

    a) Email: *jonathanneedtopoop@gmail.com*
    b) Password: UrG3nT!

    He is then sent an email for verification purposes. Upon verifying, he logs into the app successfully. His login details are then retained for future use. He now uses the search function.

2) Before he can find for a toilet, the app will ask for permission to use his live location data and only after consent is given, will access it to provide suggestions.

3) The suggested toilets come with reviews from other users that are entirely anonymous. Aside from matching his exact query with data that is publicly available, the app also provides recommendations according to the amenity use of other users (IBCF algorithm), this data will be pulled from the user-item matrix database but will not be accessed by third parties or the public.

4) After using the loo, Jonathan writes and publishes a review of the toilet. His review will be visible on the toilet profile for other users to see, but with his name censored for anonymity. His amenity use data will then be added to the user-item matrix for future recommendations for himself and other users.

Risk Assessment Matrix (RAM)

SYSTEM: FlushFinder Application			
Threat Event	Likelihood	Impact	Risk Level
Loss of Confidentiality	Likely	Moderate	**Moderate**
Loss of Integrity	Unlikely	Low	**Mild**
Loss of Availability	Likely	Low	**Mild**
		OVERALL RISK:	**Moderate**

RAM Explanations

Loss of confidentiality:

Likelihood: Likely
Despite the app not storing user location data long-term, there is still a significant likelihood of a breach occurring that could expose specific user preferences, which are non-publicly available.

Impact: Moderate
Though this data is not considered highly confidential, toilet preferences can be a very private matter to most users and hence a leak could have a moderate impact on user privacy and trust.

Loss of integrity:

Likelihood: Likely
The likelihood of loss of integrity is considered likely because malicious actors may attempt to corrupt or manipulate the app's data by submitting false reviews or ratings for toilets. Without proper data validation and authentication mechanisms, the app's data integrity can be compromised, leading to inaccurate or misleading information being presented to users.

Impact: Moderate
The impact of loss of integrity is moderate because corrupted or false data can significantly affect the app's usefulness and reliability. Users may make decisions based on inaccurate information, leading to frustration and mistrust in the app. Additionally, the presence of false reviews or ratings can undermine the app's reputation and credibility, potentially leading to a decline in user engagement and trust.

Loss of availability:

Likelihood: Likely
The app is dependent on map services like OneMap Singapore. If their services are down, so will the app in being able to locate toilets for the users.

Impact: Low
If the app becomes temporarily unavailable, the impact would be low since it provides an informational and non-critical service. Users may be inconvenienced, but it would not significantly affect the organisation or essential user activities.

Mitigation

Loss of confidentiality

Implement end-to-end encryption for all user data transmissions, including preferences. Enforce strict access controls and regular security audits to minimize unauthorized access to the data and to reveal potential weaknesses in security.. Finally, ensure prompt notification and response procedures in the event of a data breach to curb impacts on user privacy and trust.

Loss of integrity

Strong user authentication and authorization mechanisms to ensure that only verified and legitimate users can submit reviews and ratings. Implementing a moderation system can also allow for user reporting and manual review of flagged content can help identify and remove malicious or false information in a timely manner.

Loss of availability

Backup systems should be in place. This can include caching map data locally on the app, so that even if the map service is temporarily unavailable, the app can still provide basic functionality and access to cached toilet locations. Additionally, partnering with multiple map service providers and implementing a failover system can help ensure that if one provider experiences an outage, the app can seamlessly switch to an alternative provider to maintain availability.