A SOFTWARE PLATFORM FOR MONITORING PATIENTS IN SMART HOMES

RITWIK DUTTA

ARCHBISHOP MITTY HIGH SCHOOL

TALK OVERVIEW

- Related work
- SW components
- Application flow
- Frontend design
- Backend design
- Conclusions
- Future work

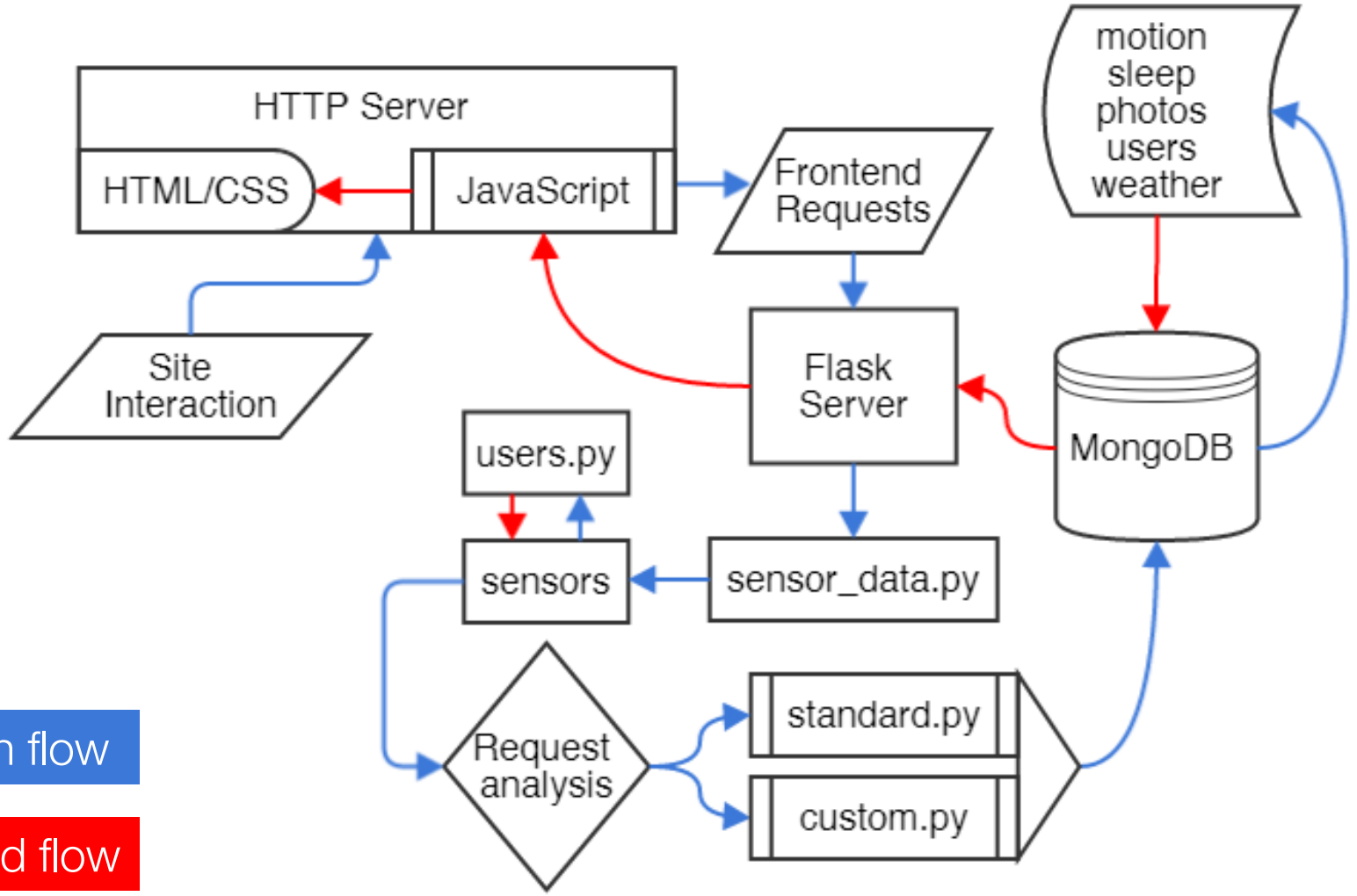
RELATED WORK

- Apple Smart Home Software
- Pegasystems Agile Healthcare Software Solutions
- Qualcomm Life 2net Platform
- Boston Software Systems
- ADT Medial Alert Monitoring
- Omnicell Unity Enterprise Platform

SW COMPONENTS

- Frontend
  - **HTTP Server:** Python SimpleHTTPServer (2.7.3)
- Backend
  - **Database:** MongoDB Portable (2.6.3)
  - **Web framework:** Flask (0.10.1)
  - **Application core:** Python (2.7.3)

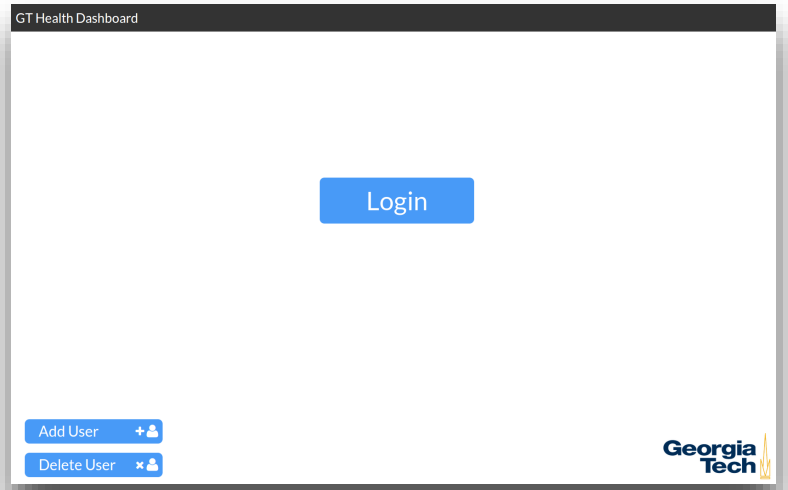
APPLICATION FLOW

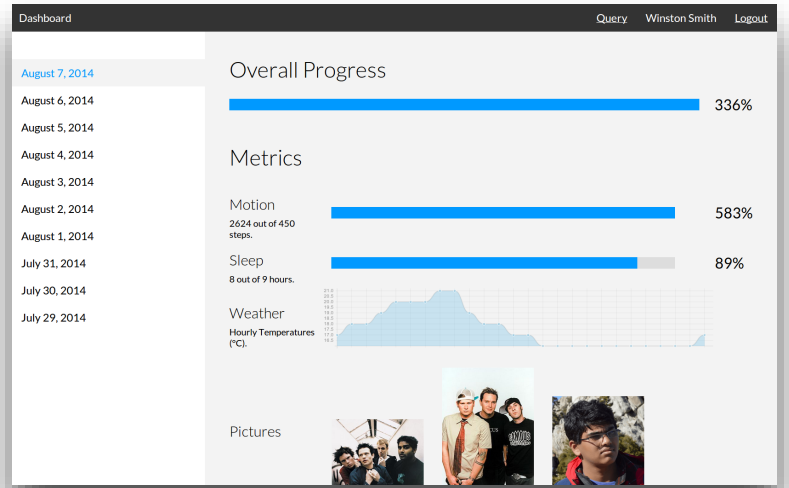


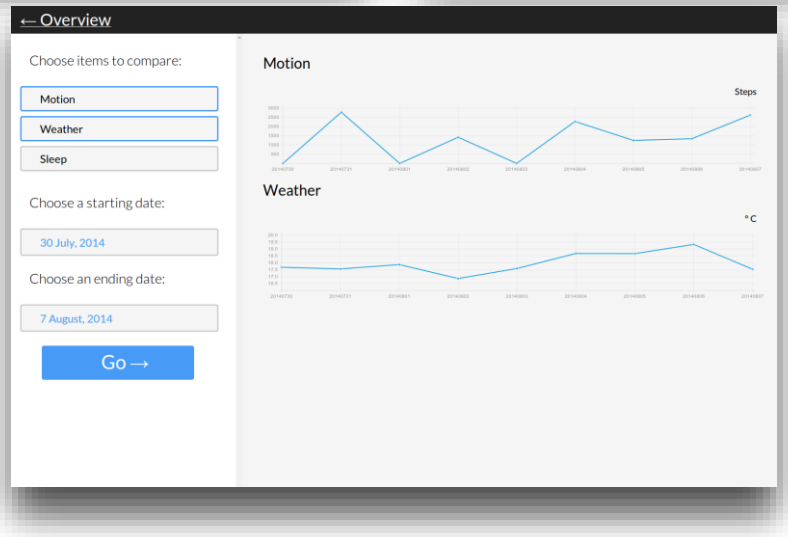
User to application flow

DB data to frontend flow

FRONTEND DESIGN



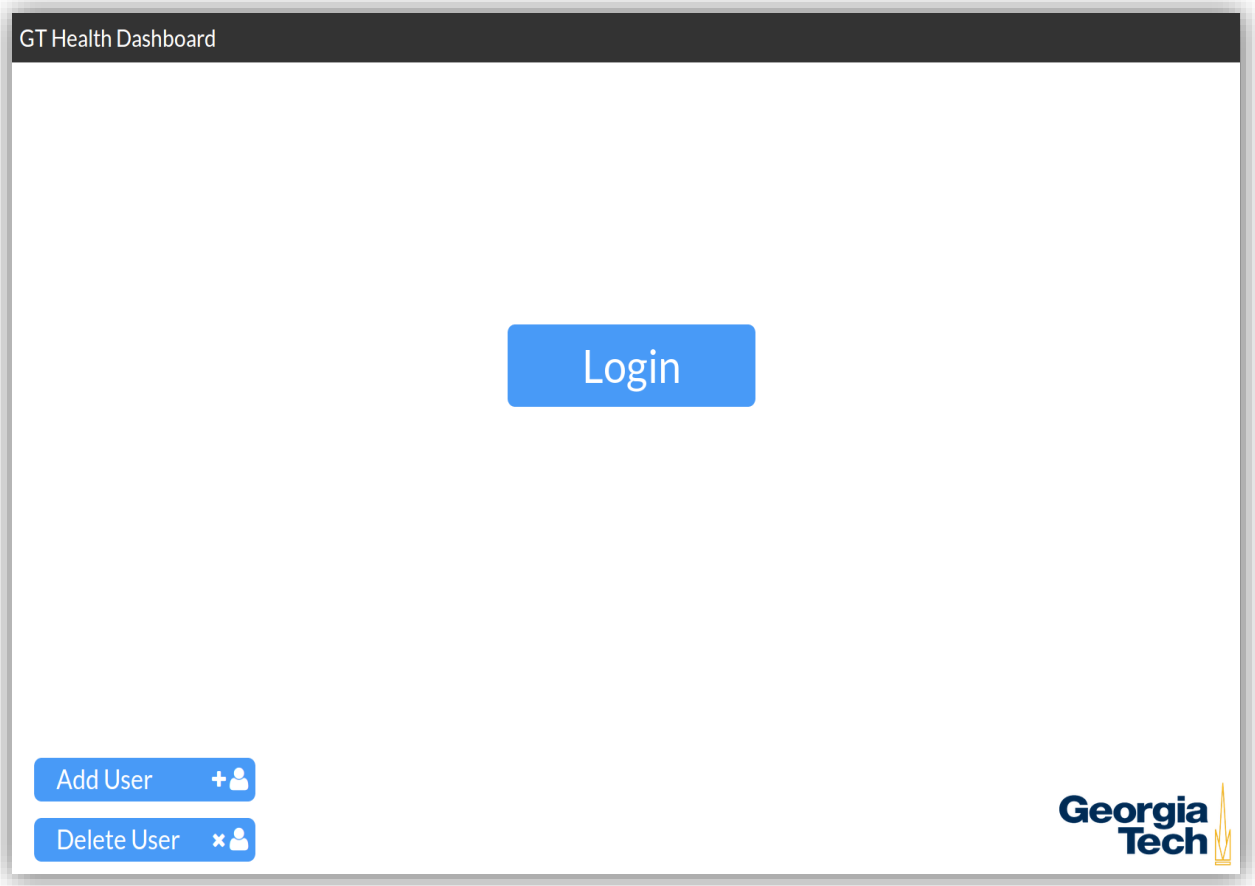




FRONTEND

- Completely custom built
- User account management
  - **Creation:** Username and metric customization
  - **Deletion**
- User account login

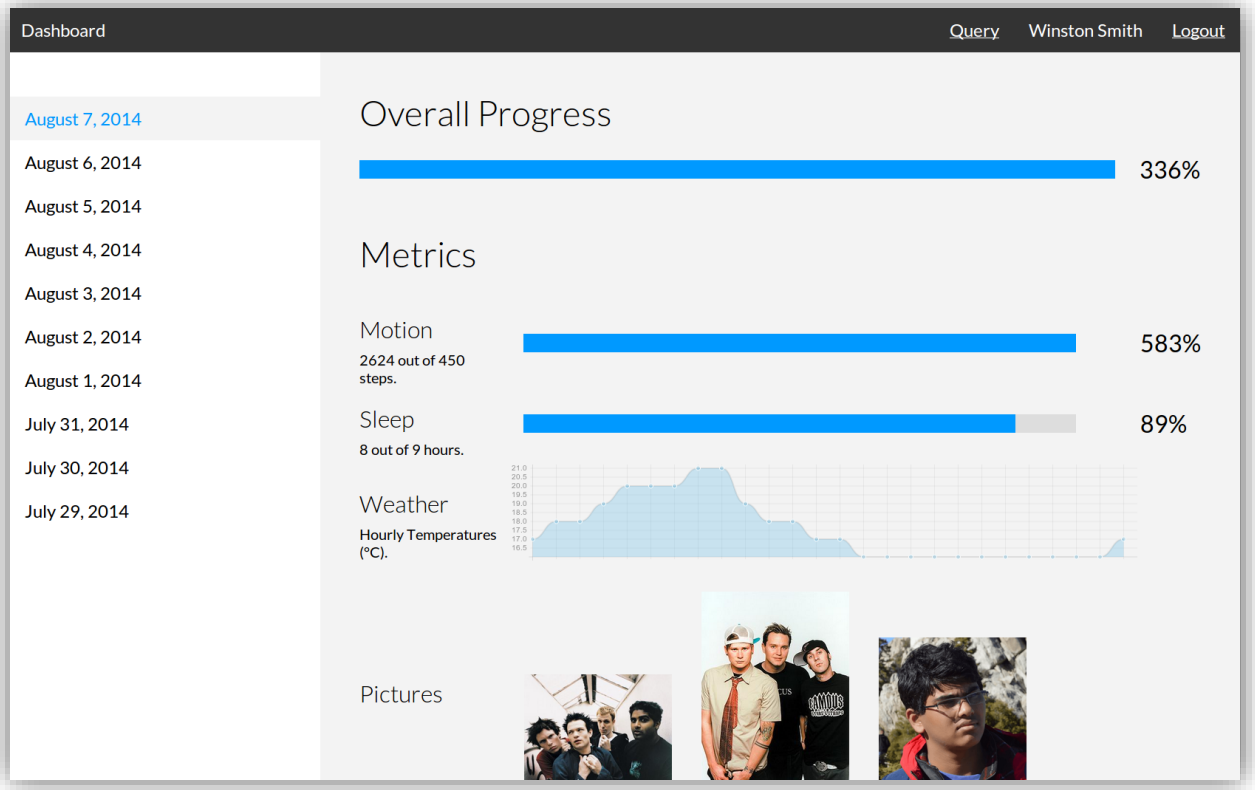
LOGIN



FRONTEND

DASHBOARD

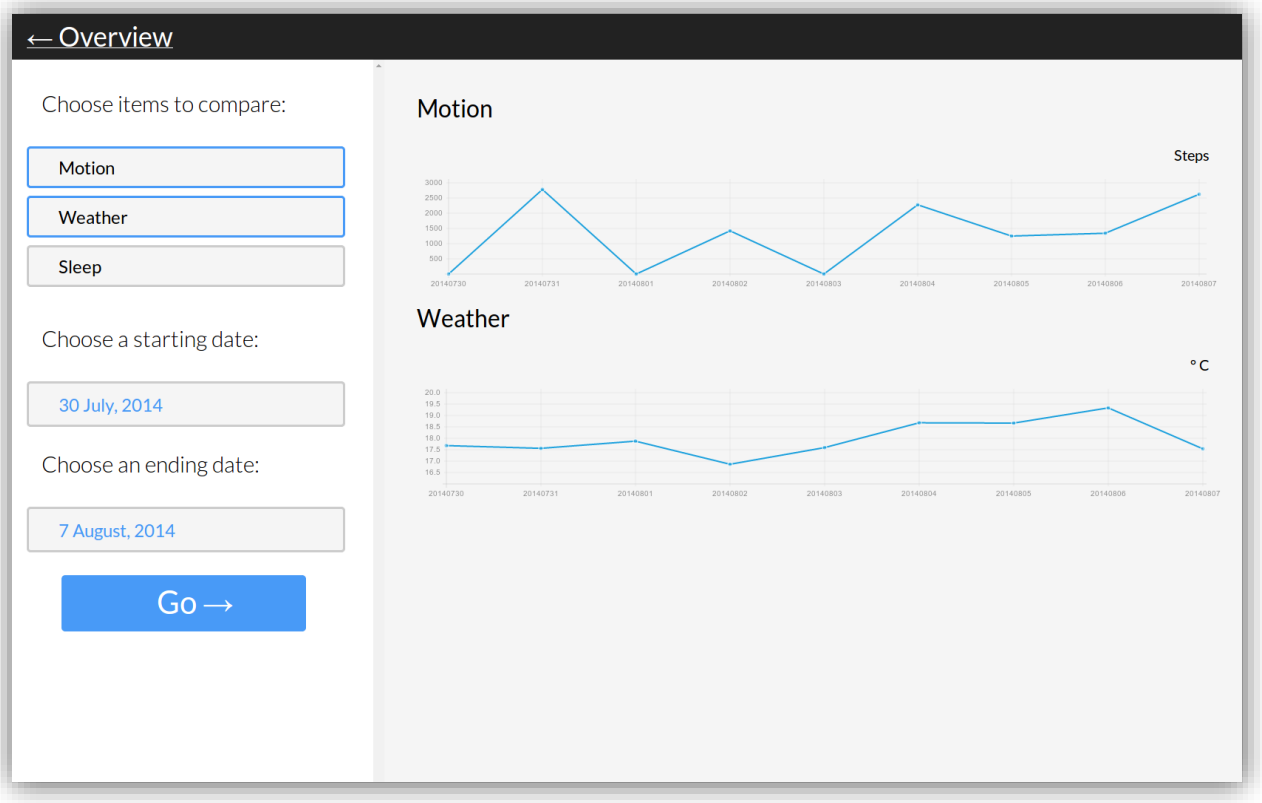
- View information by date
- Multiple data formats
  - **Percentage:** progress bar
  - **Granular data:** line graphs
  - **Pictures:** grid and lightbox
  - **Raw files:** plaintext



FRONTEND

QUERY

- Completely custom built
- Customizable data range
- Multiple graphs can be viewed at once for comparison
  - **Metric selection:** choose any number of metrics over a time range



BACKEND DESIGN

BACKEND

FLASK

- Interfaces between frontend and backend
  - **Frontend requests:** parsed and passed to backend
  - **Queries information:** pulls information from database
  - **Returns data:** formats data into readable format
  - **Frontend displays corresponding data**

BACKEND

PYTHON

- Bridges frontend and backend
  - **Flask server:** requests data from database
  - **Processing data:** pulls data from MongoDB and processes
  - **Return data:** information sent back to Flask for frontend to retrieve
- Pulls data from sensors
  - **Data logging:** sensor data is pulled and written to the database

BACKEND MONGODB

- An open source NoSQL database with JSON-like schema
- Contains collections for users and for each metric
  - **Data organization:** individual documents for each datum
  - **Sample document:**

```
{
        "date": 250110,
        "username": "wsmith1420938822",
        "metric": "sleep",
        "value": 5
}
```

  - **Query methods:** collections can be queried with multiple fields

CONCLUSIONS

- Distinguishing features of our project
  - **Licensing:** MIT licensed and freely usable and modifiable for all use
  - **Source code:** openly available and free to view, modify, or expand upon
  - **Flexibility:** easily expandable for more metrics and sensors
  - **Deployment:** simple download-and-run deployment on local or cloud platforms

FUTURE WORK (PT. 1)

- Login
  - **Password system:** more secure for patients and caretakers
  - **User account controls:** moved to admin page
- Query
  - **Data prediction:** use past data patterns to predict future trends
  - **Recommended queries:** analyze data to find “interesting” patterns to show user
  - **Data anomalies:** detect and impute anomalies in sensor data streams

FUTURE WORK (PT. 2)

- Site design
  - **Responsive:** mobile and tablet friendly design
  - **Completely dynamic pages:** use web framework for all web content generation
- Web framework
  - **Transition to Django:** more stable and mature framework for a better platform
- Project platform
  - **Transition to C/C++:** more efficient and faster data processing for project backend