1. Introduction

1.1. Problem Background

1.1.1. 57% of parents have a hard time keeping track of children's schedules, even when using digital technology.

1.1.2. Unpredictability and uncontrollability of everyday responsibilities can trigger stress.

1.1.3. Children with stressed parents are 8 times more likely to be stressed themselves.

1.1.4. Children experience less stress when they are aware of their routine or can rely on a set schedule.

1.2. Problem Description

1.2.1. Current solutions do not provide parents with a comprehensive way to plan and coordinate multiple schedules and communicate the schedules to their children.

1.2.2. Lack of organization and routine puts additional stress on children.

1.2.3. When children are not informed of their schedule, they experience uncertainty and stress.

1.3. Solution Description

1.3.1. Continuously monitor for new information from the family’s online calendars and task lists.

1.3.2. Assemble and present child-specific information in an age-appropriate format for each child.

1.3.3. Provide children with consistent information about their daily routines that are aligned with the family’s schedules.

2. Kiddy-Up Product Description

2.1. Key Product Features and Capabilities

2.1.1. The parent can register for an account and add their children to Kiddy-Up

2.1.2. Parent can add their calendars, task lists, and other information sources to Kiddy-Up

2.1.3. Kiddy-Up Analyzes data from online sources to find relevant information for the children

2.1.4. Kiddy-Up formats the relevant information for the child’s comprehension level and presents to the child at the right date/time/condition

2.1.5. Kiddy-Up tracks tasks/events completion status and reports back to the parent, allowing parental validation if necessary.

2.1.6. Kiddy-Up tracks reward goals and progress for each child as they earn points toward their chosen goals by completing events and tasks and
presents for the children and parents

2.2. Major Components (Hardware/Software)

2.2.1. Hardware

2.2.1.1. Any device with browser-based web interface
2.2.1.2. Any Android device with web connectivity
2.2.1.3. Any iOS device with web connectivity
2.2.1.4. AWS Serverless infrastructure
   2.2.1.4.1. API Gateway: Make web calls to other services (get calendar/task items)
   2.2.1.4.2. Lambda: Run code w/o servers
   2.2.1.4.3. S3 (Simple Storage Service): Object Storage
   2.2.1.4.4. Aurora: AWS Relational Cloud database
   2.2.1.4.5. AWS Comprehend: Natural Language Processing, Identify entities in written information (events/tasks)

2.2.2. Software

2.2.2.1. Web Programming: HTML5, CSS3, JS
2.2.2.2. Mobile Programming: HTML5, CSS3, JS, Python
2.2.2.3. Lambda Functions: Python
2.2.2.4. Database: Aurora Serverless
2.2.2.5. IDE: Eclipse, Kivy
2.2.2.6. Code Repository (Version control, issue tracking): Gitlab
2.2.2.7. Group Collaboration: Slack, Google Hangouts

3. Identification of Case Study

3.1. Who is Kiddy-Up for?

3.1.1. Parents/Caregivers/etc
   3.1.1.1. Busy parents
   3.1.1.2. Have Internet access
   3.1.1.3. Have online calendars and tasklists accessible via API calls

3.1.2. Children
   3.1.2.1. Primarily school-age (5-12)
   3.1.2.2. Kids that do not have their own online calendars/tasklists
   3.1.2.3. Kids that can use technology (mobile device or computer)

3.2. What will it be used for?

3.2.1. Organizing Families so all are on the same page
3.2.2. Informing Children of what is happening in their lives/families
3.2.3. Reducing the burden on the parent to keep children aware

3.3. Who else might benefit?

3.3.1. Teachers/Schools
3.3.2. Clubs and groups

4. Kiddy-Up Product Prototype Description

4.1. Prototype Architecture (Hardware/Software)

4.1.1. Hardware
   4.1.1.1. Any device with browser-based web interface
4.1.2. Software
   4.1.2.1. Docker (LAMP server)
      4.1.2.1.1. Apache
      4.1.2.1.2. MySQL DB
      4.1.2.1.3. Python
   4.1.2.2. Web Programming: HTML5, CSS3, JS, Django
   4.1.2.3. Python (SpaCy - Python Natural Language Libraries, Django)
   4.1.2.4. IDE: Eclipse
   4.1.2.5. Code Repository (Version control): GitLab
   4.1.2.6. Project Management, issue tracking: RedMine

4.2. Prototype Features and Capabilities
   4.2.1. Parent can add their calendars, task lists, and other information sources to Kiddy-Up
   4.2.2. Kiddy-Up Analyzes data from online sources to find relevant information for the children
   4.2.3. Kiddy-Up formats the relevant information for the child’s comprehension level and presents to the child at the right date/time/condition
   4.2.4. Kiddy-Up tracks tasks/events completion status and reports back to the parent.

4.3. Prototype Development Challenges
   4.3.1. Obtaining the AWS services - ODU Restriction
   4.3.2. Database configuration
   4.3.3. Automatic importation from third party sources.
   4.3.4. Identify and handle event/task conflicts
   4.3.5. Train/Use/Learn SpaCy to process imported data
   4.3.6. Convert task/events into grade-appropriate elements

5. Glossary

Apache: A free and open-source cross-platform web server software.

API (Application Programming Interface): A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

AWS (Amazon Web Services): A cloud services platform that offers compute power, database storage, content delivery, and other functionality.
**API Gateway**: A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

**Aurora**: A MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.

**Aurora Serverless**: An on-demand, auto-scaling configuration for Amazon Aurora.

**Comprehend**: A natural language processing (NLP) service that uses machine learning to find insights and relationships in text. No machine learning experience is required.

**Django**: A Python-based free and open-source web framework.

**Docker**: A set of platform-as-a-service products that use OS-level virtualization to deliver software in packages called containers.

**Event**: A single occurrence that will not happen on a regular basis. An example of an event would be a science fair on Saturday at 10:00 am.

**Lambda**: A serverless compute service that runs code in response to events and automatically manages the underlying compute resources.

**MySQL**: An open-source relational database management system.

**Online Sources**: The Internet based calendars and task lists that parents are currently using to coordinate their families. These are also the sources that Kiddy-up will draw from.

**Redmine**: An online project management tool that offers services such as issue tracking and a
Gantt chart.

**S3 (Simple Storage Service):** An object storage service that offers industry-leading scalability, data availability, security, and performance.

**SpaCy:** An open-source software library for advanced natural language processing.

**Task:** A reoccurring chore that the child must complete. An example of a task would be taking out the trash.

### 6. References


https://www.webmd.com/balance/guide/causes-of-stress#1


https://www.klaschools.com/importance-of-routine-for-children/

