

成果発表

森脇佑介 楠見隆紘
今城慧彦 東田紫門

1.概要

システムの概要及び機能

- 人感センサーの反応に応じてユーザーの通知、操作
- ユーザーはLINEの通知からカメラやスピーカーを操作可能
- ユーザーはOneDriveの共有ファイルで画像の確認が可能



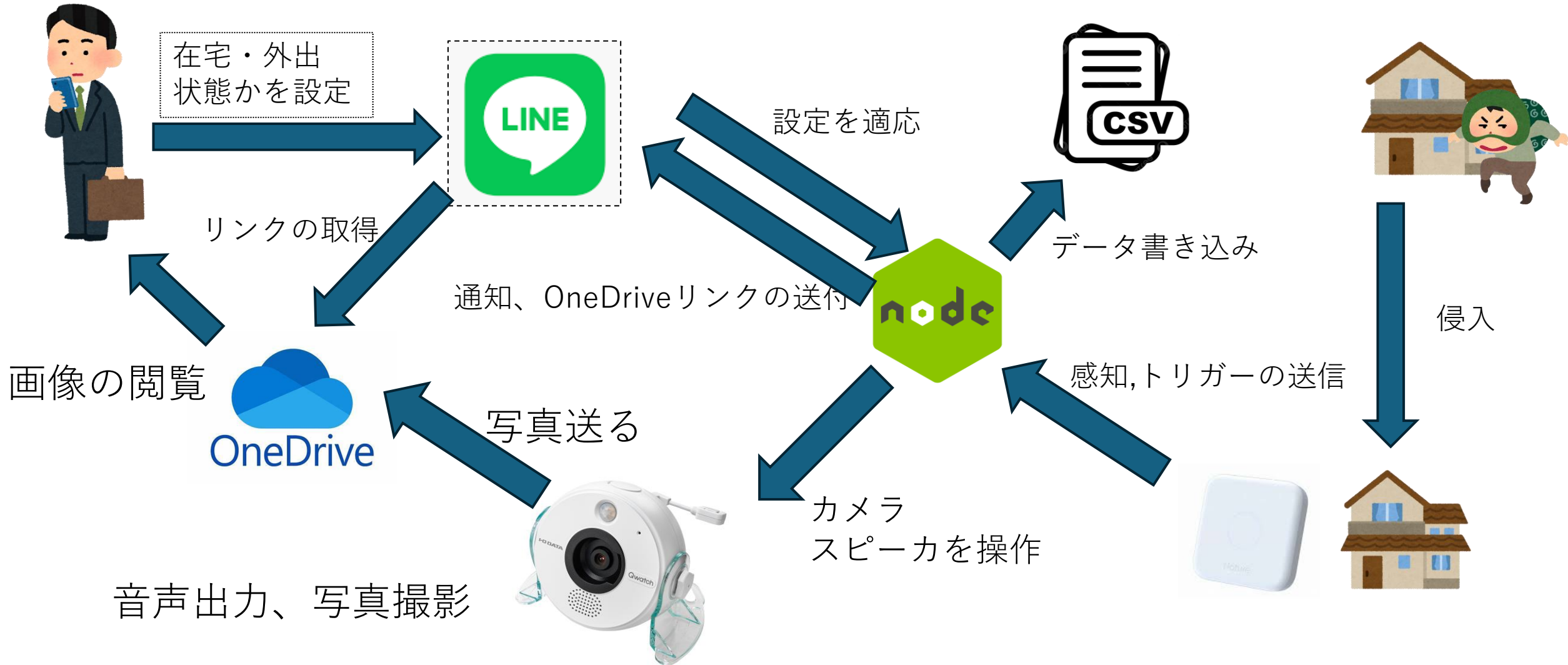
想定する利用者

- ・ 住居を持つ**すべての人**が対象ではあるが、特に家のセキュリティや周囲の治安に**不安を持つ人**や、1人暮らしの人にとって有用なシステムであると考ええる。例えば、**一人暮らしの女性**などの防犯上のトラブルを防ぐ事を期待できる。

（・ 研究室の警備？）

2.設計

システム処理の流れ

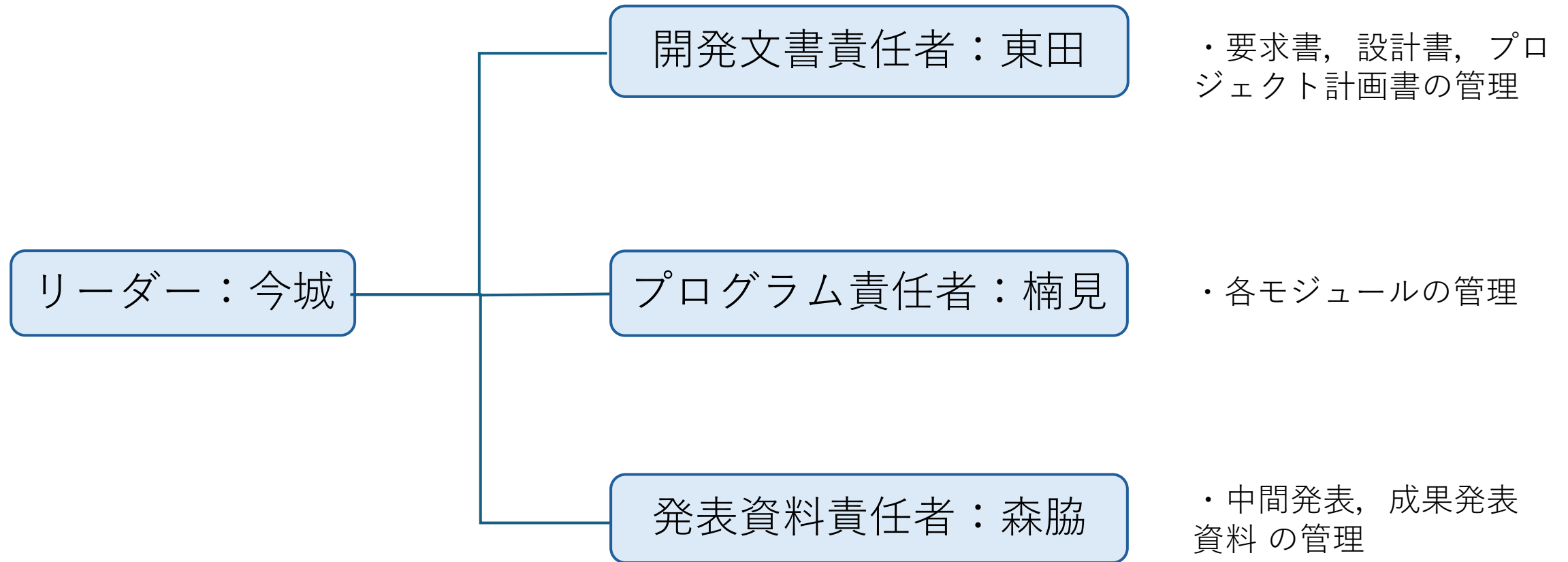


開発したモジュール

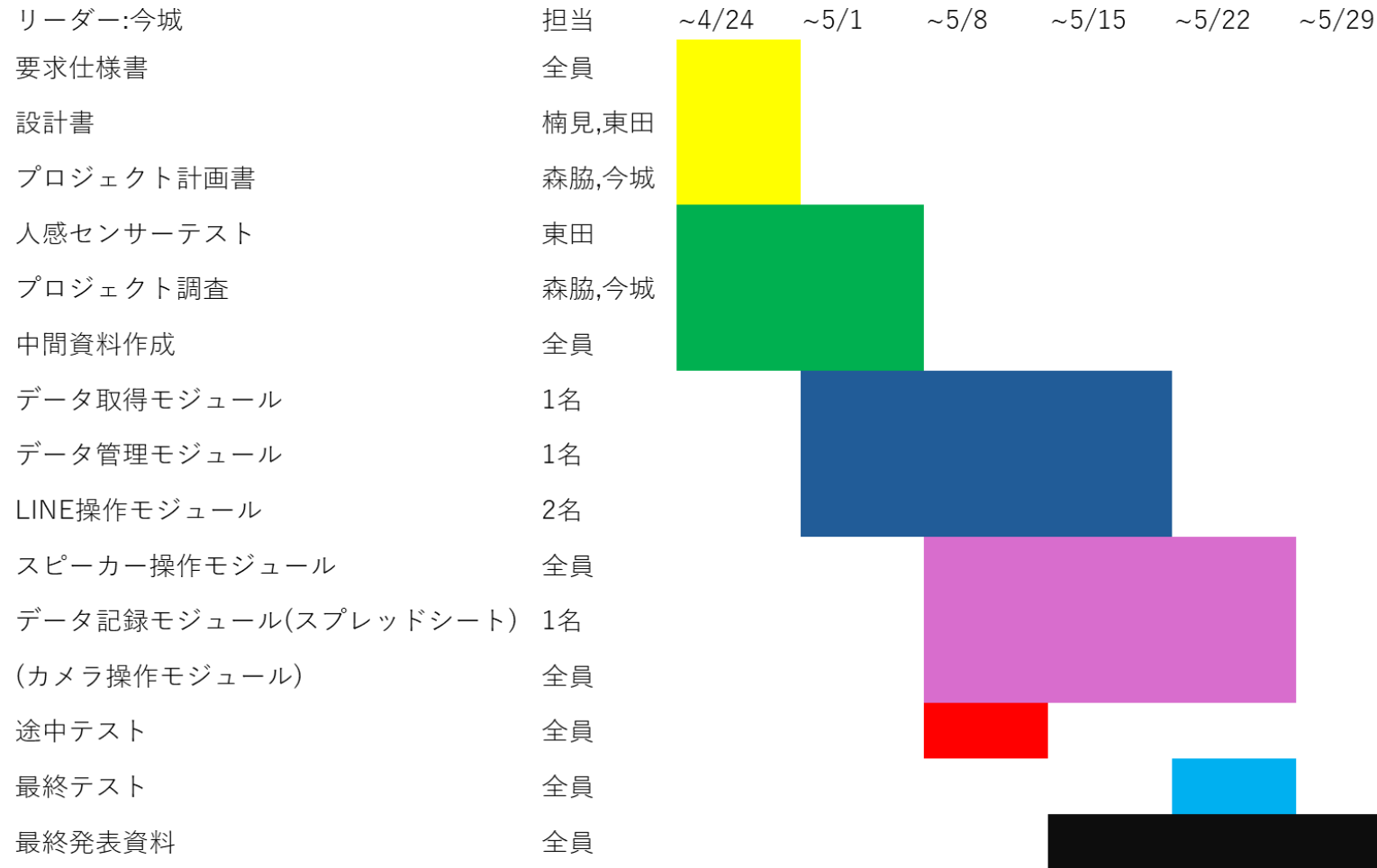
- カメラ操作モジュール
- データ取得モジュール
- データ管理モジュール
- LINE操作モジュール
- スピーカ操作モジュール
- データ記録モジュール

3.プロジェクト計画

開発体制



開発スケジュール



要求仕様

ユーザーは、LINEによって帰宅か外出中かを選択することができる

外出中に人感センサーが反応した際、ユーザーのLINEに通知が届く。

ユーザーは通知に応じてスピーカーで警告音を鳴らすかどうか選択する。

カメラを起動して対象を撮影し、LINEにて画像をユーザーに送信する

ユーザーは、検知された情報を後から参照できる。

要求仕様と設計仕様の達成度

達成

ユーザーは、LINEによって帰宅か外出中かを選択することができる。
外出中に人感センサーが反応した際、カメラで対象を撮り、スピーカーで警告音を鳴らす。

未達成

カメラを起動して対象を撮影し、LINEを用いて画像をユーザーに送信する。

ユーザーは、検知された情報を後から参照できる。

部分的な達成

カメラを起動して対象を撮影する。(LINEでの送信の実装は未達成)
ユーザーは、検知された情報を後から参照できる(スプレッドシートでの参照は可能)。

仕様変更

スピーカーの警告音再生の実行を手動ではなく自動で実行する仕様に変更にした。

各モジュール

カメラ関連処理 `function downloadImage()`

カメラを起動し画像を撮影し,指定したディレクトリ内に保存するとともに,スピーカーから警告音声を流す.カメラの操作はプログラムからコマンドラインを実行することにより実装した.

LINEのメッセージ判定 `const handleEvents = events =>`

LINEに送られたチャットがmessageかどうかを判定する。
Message以外の画像などが送られてきた場合はUnknown event type: `${event.type}``とチャットし、そうでなければhandleMessageEvent()に移行する。

メッセージの文字判定 `const handleMessageEvent = event =>`

「外出」、「在宅」、それ以外のメッセージの条件分岐をする
外出の場合はhandleGoOut()に、在宅の場合はhandleStayAtHome()にその他のメッセージはhandleOtherMessages()に移行する。

各モジュール

状態管理モジュール

```
const handleStayAtHome = event =>
```

```
const handleGoOut = event =>
```

ユーザーから送信されたLINEのメッセージに応じてそれぞれ外出,在宅の状態を管理する.

その他メッセージ管理モジュール

```
const handleOtherMessages = event =>
```

ユーザーから送信された在宅,外出処理を行う以外のLINEのメッセージに対して定型文を返信する

```
messages: [{ type: "text", text: "外出または在宅をチャットしてください"}]
```

各モジュール

センサー情報取得モジュール

Const getSensorData = async () =>

Remo3の人感センサーデータの情報を取得し,csvファイルに記入する.

csvファイル記入モジュール

const appendToCSV = (executionTime, reactionTime) =>

getSensorDataで記入した情報に基づいて人感センサーが反応した時間をcsvファイルに記入する

開発計画進行の評価

カメラモジュール

予定通りに完成.いくつか問題はあったものの順調に進行,完成

LINEモジュール

LINE自体が外部のサーバで動かすことを想定しているので、それをオフライン環境で実装することに苦勞した。

苦勞した点

- ・カメラのAPI仕様書の記述が一部間違っていた.
- ・カメラの認証の関係上,プログラムをローカル環境で動かす必要があったため,複雑な環境が必要になり,OSを跨ぐ開発環境が原因で一部の機能が実装できなかった.
- ・上記の理由により途中から開発環境が変わり開発が混乱した.

4.デモンストレーション

Special Thanks:吉田教授
(声の出演)