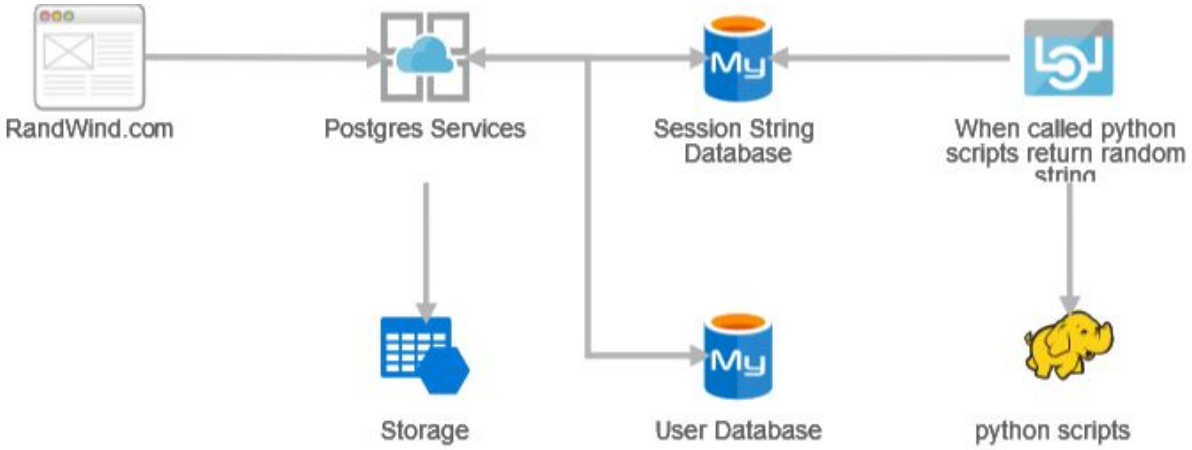
1. **Feature List**
    1. Track and calculate the change in fluid motion using visual processing
        a. This should be truly random
    2. Generate random numbers from the inputs of the fluid motion
        a. Allow users to request a certain number of random numbers (binary, 0-9, ASCII, etc.) and deliver them as a set to users.
    3. Website for users to interact with
        a. This will allow the user to request and receive the random numbers in form view.
    4. All databases used are encrypted
        a. This is to allow users to use the randomly generated numbers for secure things like passwords or their own encryption
    5. User login and authentication
        a. Allows each user to store requested number set with security measures.

2. **Architecture Diagram**

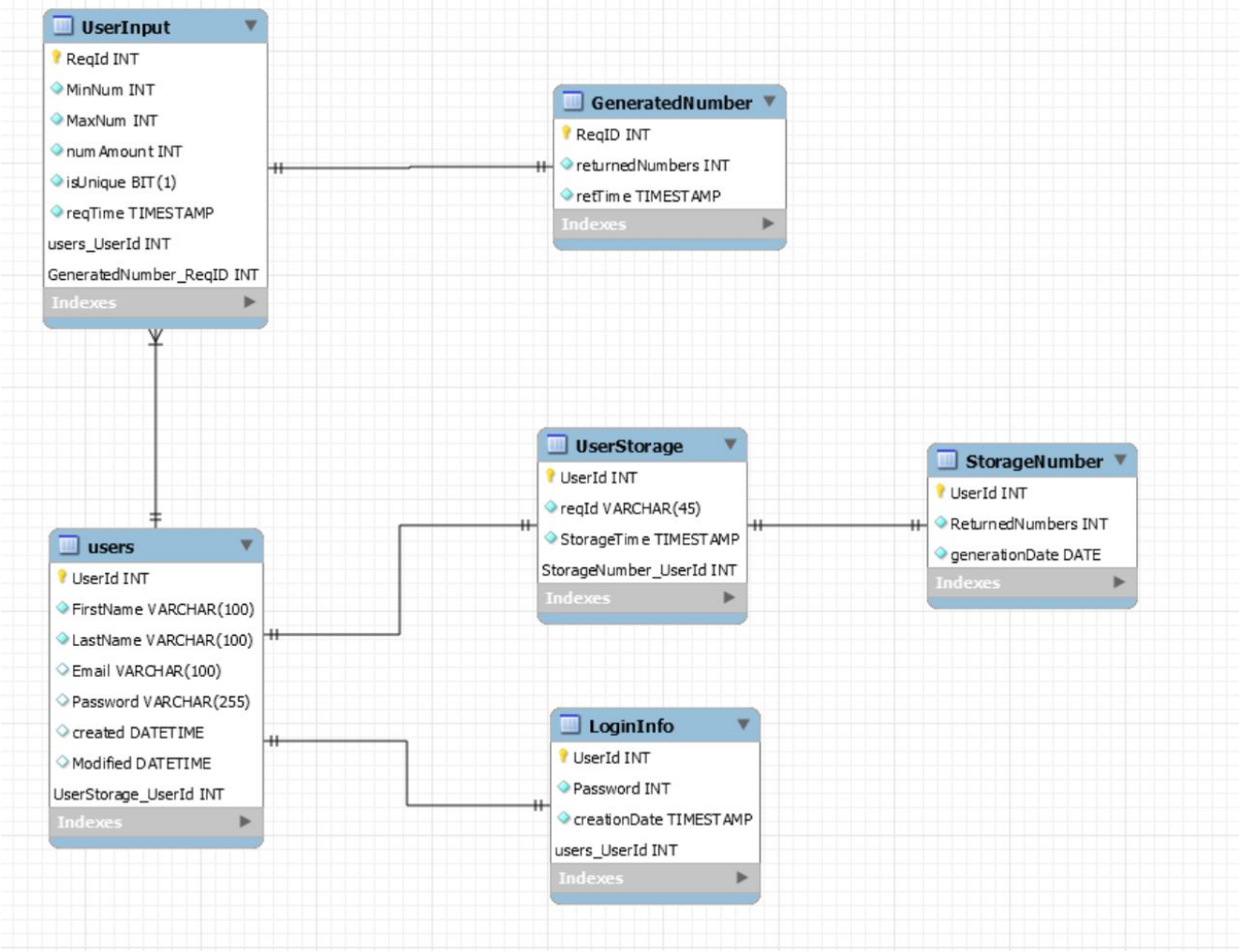


3. **Front End Design**
We created a simple webpage already. It can be found at:
https://luke-favret.github.io/RandWind/RandWind_Web/views/home.html

4. **Web Service Design**
We are currently not using any APIs.

5. **Database design**
Summary: Our database will manage 3 things: the user's input for each request, the generated numbers returned by the python script, user management/authentication, and stored user generations.

The request and return request tables must be scrubbed immediately after the information is passed to the next “place.”




```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema mydb
```

```
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----------------------------------------------------
-- Table `mydb`.`StorageNumber`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`StorageNumber` (
  `UserId` INT NOT NULL,
  `ReturnedNumbers` INT NOT NULL,
  `generationDate` DATE NOT NULL,
  PRIMARY KEY (`UserId`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`UserStorage`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`UserStorage` (
  `UserId` INT NOT NULL,
  `reqId` VARCHAR(45) NOT NULL,
  `StorageTime` TIMESTAMP NOT NULL,
  `StorageNumber_UserId` INT NOT NULL,
  PRIMARY KEY (`UserId`, `StorageNumber_UserId`),
  INDEX `fk_UserStorage_StorageNumber1_idx` (`StorageNumber_UserId` ASC) VISIBLE,
  CONSTRAINT `fk_UserStorage_StorageNumber1`
    FOREIGN KEY (`StorageNumber_UserId`)
    REFERENCES `mydb`.`StorageNumber` (`UserId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`users`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`users` (
  `UserId` INT NOT NULL AUTO_INCREMENT,
  `FirstName` VARCHAR(100) NOT NULL,
  `LastName` VARCHAR(100) NOT NULL,
  `Email` VARCHAR(100) NULL,
  `Password` VARCHAR(255) NULL,
  `created` DATETIME NULL,
```

```
  `Modified` DATETIME NULL,
  `UserStorage_UserId` INT NOT NULL,
  PRIMARY KEY (`UserId`, `UserStorage_UserId`),
  INDEX `fk_users_UserStorage1_idx` (`UserStorage_UserId` ASC) VISIBLE,
  CONSTRAINT `fk_users_UserStorage1`
    FOREIGN KEY (`UserStorage_UserId`)
    REFERENCES `mydb`.`UserStorage` (`UserId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`GeneratedNumber`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`GeneratedNumber` (
  `ReqID` INT NOT NULL,
  `returnedNumbers` INT NOT NULL,
  `retTime` TIMESTAMP NOT NULL,
  PRIMARY KEY (`ReqID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`UserInput`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`UserInput` (
  `ReqId` INT NOT NULL,
  `MinNum` INT NOT NULL,
  `MaxNum` INT NOT NULL,
  `numAmount` INT NOT NULL,
 `isUnique` BIT(1) NOT NULL,
  `reqTime` TIMESTAMP NOT NULL,
  `users_UserId` INT NOT NULL,
  `GeneratedNumber_ReqID` INT NOT NULL,
  PRIMARY KEY (`ReqId`, `users_UserId`, `GeneratedNumber_ReqID`),
  INDEX `fk_UserInput_users1_idx` (`users_UserId` ASC) VISIBLE,
  INDEX `fk_UserInput_GeneratedNumber1_idx` (`GeneratedNumber_ReqID` ASC) VISIBLE,
  CONSTRAINT `fk_UserInput_users1`
    FOREIGN KEY (`users_UserId`)
    REFERENCES `mydb`.`users` (`UserId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_UserInput_GeneratedNumber1`
    FOREIGN KEY (`GeneratedNumber_ReqID`)
    REFERENCES `mydb`.`GeneratedNumber` (`ReqID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`LoginInfo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`LoginInfo` (
  `UserId` INT NOT NULL,
  `Password` INT NOT NULL,
  `creationDate` TIMESTAMP NOT NULL,
  `users_UserId` INT NOT NULL,
  PRIMARY KEY (`UserId`, `users_UserId`),
  INDEX `fk_LoginInfo_users_idx` (`users_UserId` ASC) VISIBLE,
  CONSTRAINT `fk_LoginInfo_users`
    FOREIGN KEY (`users_UserId`)
    REFERENCES `mydb`.`users` (`UserId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```