

2024-2 오픈소스 소프트웨어 프로젝트 최종발표

ShiftMate

아르바이트 일정 배정 시스템

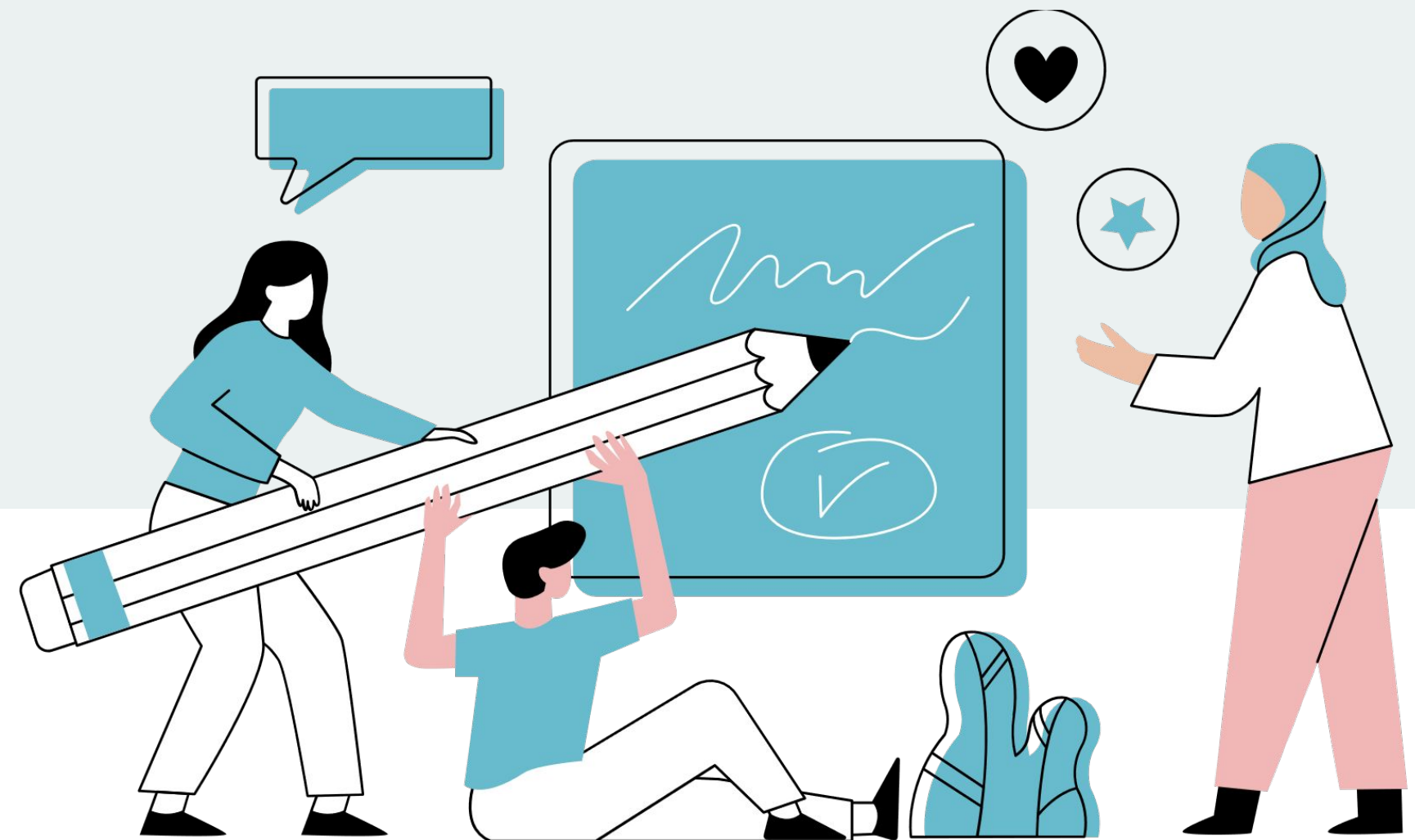
03조 StackOverflows

2021110806 김민정

2021112379 김서연

2020113296 윤석규

2020111670 지경민



목차

ShiftMate개요 다이어그램 기능설명 시연 기대효과 및 확장성 추후 계획 QnA

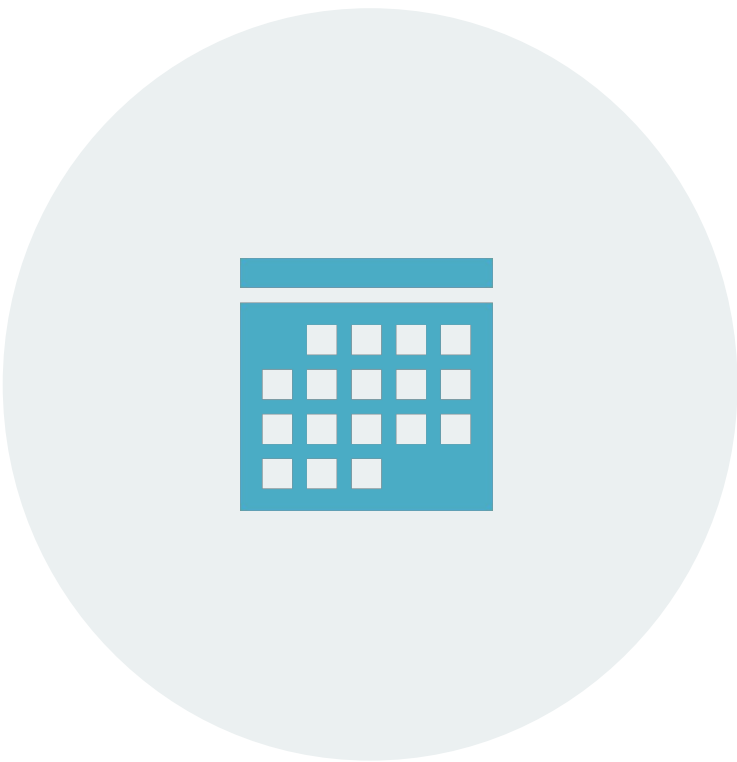
CONTENTS

1. ShiftMate 개요
2. 다이어그램
3. 기능 설명
4. 시연
5. 기대효과 및 확장성
6. 추후 계획
7. QnA

프로젝트 개요



근무 일정 관리 시스템의 부재



근무 일정 생성 알고리즘



편의성 및 공정성



근무자가 같은 시급을 받고 같은 업무를 하는 업장



업무 단위가 아닌 시간 단위로 근무를 하는 업장



직급 또는 숙련도에 따라 근무 배정이 달라지지 않는 업장

최종 설계 목표

ShiftMate개요 다이어그램 기능설명 시연 기대효과 및 확장성 추후 계획 QnA



근무 일정 배정 자동화

근무자의 선호도를 반영한
근무 일정 배정을 통해
시간절약 및 비용절감

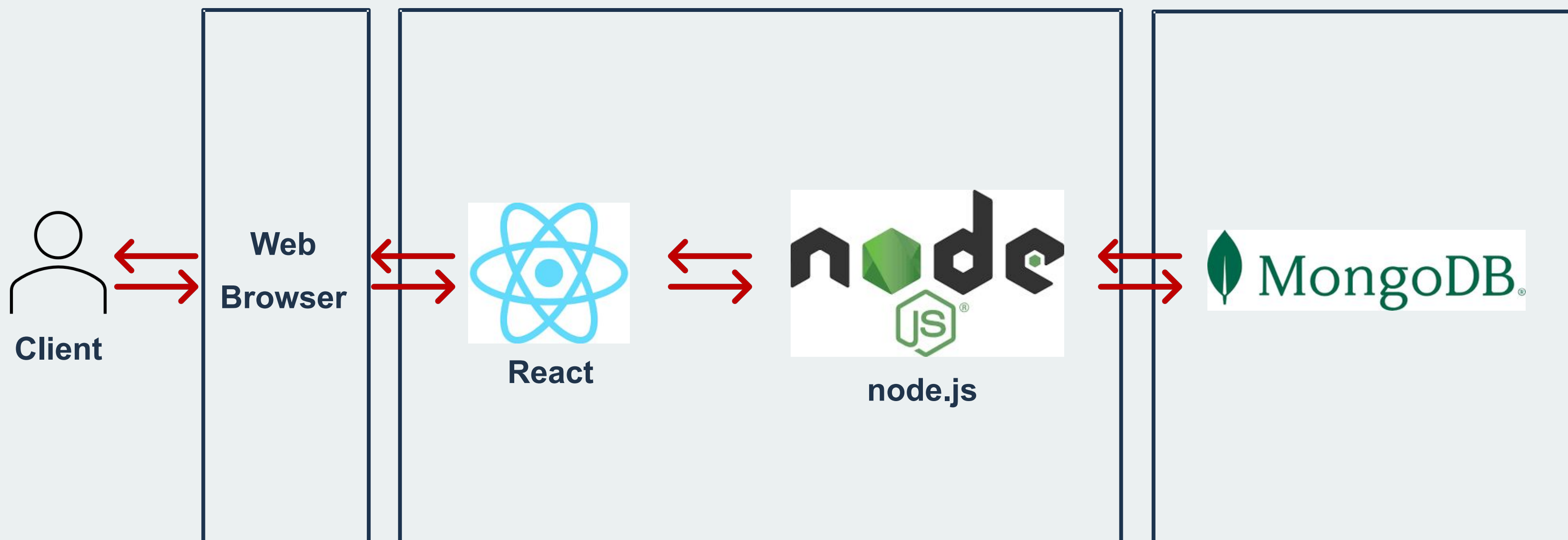


근무자, 관리자 모두에게 친화적인 일정 관리 시스템

근무자 - 관리자 간 일정 관리를
시스템을 통해 자동화해
소통부재, 일정 불일치 문제 해결

Architecture

[ShiftMate개요](#) [다이어그램](#) [기능설명](#) [시연](#) [기대효과 및 확장성](#) [추후 계획](#) [QnA](#)



흐름도

1.[관리자]근무 생성

관리자가 근무시간, 마감기한,
근무인원을 추가하여 근무표 생성

2.[근무자]근무신청

해당 근무표를 본 근무 희망자들이
해당시간에 근무를 신청

3.근무자 선발

마감기한이 지나면 알고리즘을
활용해 근무자를 선발
마지막 근무시간이 오래된
사람일수록, 근무신청 거절횟수가
많은 사람일수록 선발률 상승

4.(근무자)근무표 조회 (관리자)근무표 수정

근무자 개인은 신청결과를 확인
일정에 변동이 생긴경우,
관리자는 근무를 조회, 수정할 수
있음

Fair queuing + Aging algorithm

1. 지원자 점수 계산

- 최근 근무한 시간 + 마지막 근무 일자를 기준
- 오래 일하지 않은 사람에게 더 높은 점수를 부여
- 배정에서 여러 번 밀려난 지원자에게 추가 점수를 부여

2. 근무자 배정

- 높은 점수순으로 선발
- 선발자 중 결원이 생길 경우 거절된 사람중 높은 점수를 받은 사람부터 선발 되는 구조 추가

3. 미배정 지원자 관리

- 최종 거절된 근무자들은 다음 근무신청시 높은 점수를 받을 확률이 높도록 설정

시연

기대효과 및 확장성

ShiftMate개요 다이어그램 기능설명 시연 기대효과 및 확장성 추후 계획 QnA



관리자-근무자간 정보불일치 해소

조직 전체에 투명성을 제공하여 정보불일치를 해소함

근무배정에 공정하고 유연하게 접근하고, 더 나은 이해와 협업을 도모



정오

실시간 협업 도구를 적극 활용하여 신속한 소통과 의사 결정 프로세스를 강조하여 팀 간 협력을

강화합니다.

이후 시스템 개선 및 기능 확장 제안

[ShiftMate개요](#) [다이어그램](#) [기능설명](#) [시연](#) [기대효과 및 확장성](#) [추후 계획](#) [QnA](#)



알고리즘 실행 날짜 및 시간 지정 기능



알고리즘이 배정한 근무표가 조회 페이지에 게시되면 근무자에게 알림을 전송하는 기능



두 개 이상의 업장의 근무를 신청하고 조회하는 기능



QnA

감사합니
다

부록

1. ShiftMate 개요

연락처 안내

더 궁금한 점은 아래 연락처와 홈페이지를
참고해주세요.



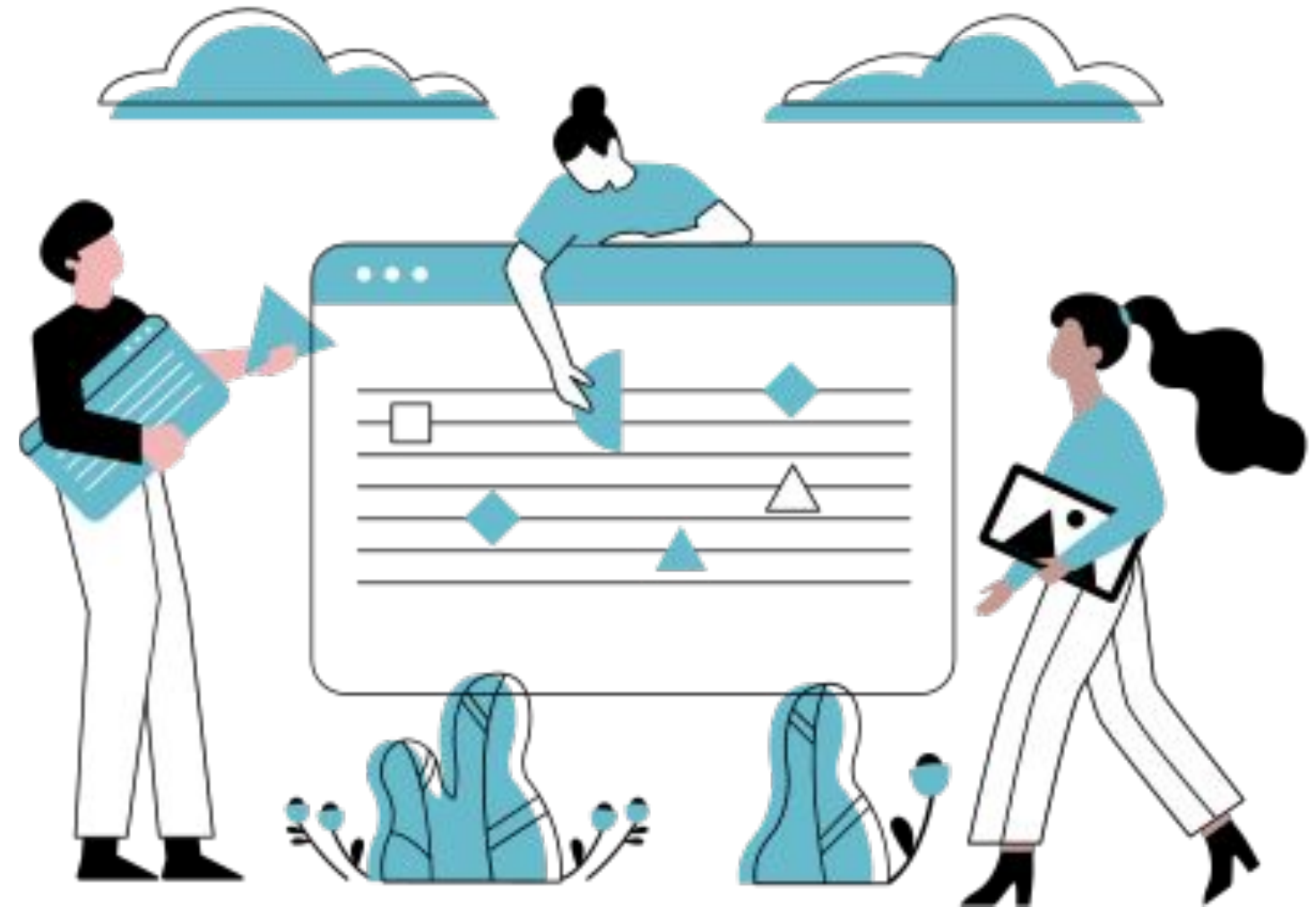
이메일

hello@reallygreatsite.com



웹사이트

www.reallygreatsite.com



서비스 특징 4가지

개인정보 보호 강화

최고 수준의 보안 프로토콜을 도입하여 사용자 데이터와 개인정보를 철저히 보호하며, 안전한 서비스 제공

AI와 머신러닝 적용

인공지능과 머신러닝 기술을 활용하여 개인화된 서비스를 제공하고, 사용자의 행동 패턴에 대한 통찰력을 확장



혁신적 기술 구현

최신 IT 기술을 적용하여 사용자 경험을 혁신적으로 개선하고, 업계 표준을 뛰어넘는 서비스 제공

탄력적이고 확장 가능한 아키텍처

탄력적인 시스템 아키텍처를 통해 서비스 확장성을 극대화하고, 사용자 수요에 신속하게 대응

부록4. 간트차트

[illegible]

기대효과 및 확장성

부록5. ShiftMate 근무자 배정 알고리즘

Aging Algorithm

- 시스템에서 기다린 시간에 비례해서 우선순위를 부여해주는 방식
- 특정 프로세스의 우선순위가 낮아서 무한정 대기하는 경우 (starvation)를 방지
=> 상대적으로 일한지 오랜 시간이 흐른 아르바이트생에게 높은 우선순위 부여!
알고리즘 복잡도 : $O(n)$

Fair Queuing Algorithm

- 한정된 자원이 주어졌을 때, 공정하게 자원을 배분하는 것에 중점
- 가중치에 따라 공정하게 자원을 할당(가중치 = 과거에 밀려난 횟수나 최근에 일한 빈도에 따라 결정)
=> 특정 근무자가 자주 배정되는 혹은 되지 않는 문제를 방지, 근무 기회가 필요한 근무자에게
알고리즘 복잡도 : $O(\log(n))$
우선적인 기회부여

부록5. ShiftMate 근무자 배정 알고리즘

1) Aging

대량의 높은 우선순위를 가진 자원이 한번에 발생할 경우, 상대적으로 낮은 우선순위를 가진 자원은 ‘기아상태’(무한정 대기)에 빠질 수 있음.

2) Fair queuing

한정된 자원이 주어졌을 때, 공정하게 자원을 배분하는 방식. Fair queuing 알고리즘은 여러 자원이나 작업을 균등하게 배정하는 데 중점을 둠. 가중치가 높으면 더 많은 자원을 할당하여 프로세스를 빨리 끝내게끔 함. (즉, 우선순위가 높아지면 더 높은 점수를 얻도록 만드는 방식) 이 알고리즘 적용시, 결과적으로 모든 요청이 공정하게 대기열에 추가되고 처리되도록 함(특정 자원이 지속적으로 대기열에서 밀려나지 않도록 하는 방식)

3) Greedy algorithm

각 상황에서 최대한의 수행능력, 최댓값을 갖도록 선택하는 알고리즘(그림 첨부) 즉, 잘하는 아르바이트생에게 더 높은 점수를 부여하여 선호 시간 근무에 우선권을 주는 방식.

개발 현황 : Backend

알고리즘 개발:결과

```
{
  "message": "근무자 선발 완료",
  "selectedWorkers": [
    {
      "name": "최지우",
      "score": 355.7309661111111
    },
    {
      "name": "정우성",
      "score": 332.7309661111111
    },
    {
      "name": "박민수",
      "score": 305.7309661111111
    }
  ],
  "rejectedWorkers": [
    {
      "name": "김철수",
      "score": 283.7309661111111
    },
    {
      "name": "이영희",
      "score": 236.73096611111112
    }
  ],
}
```



```
{
  "_id": "67330610c896927f8ab7dd1e",
  "workerId": null,
  "name": "박민수",
  "start": "2024-11-12T09:00:00.000Z",
  "end": "2024-11-12T12:00:00.000Z",
  "lastShiftStart": "2024-10-30T10:00:00.000Z",
  "lastShiftEnd": "2024-10-30T14:00:00.000Z",
  "status": "Approved",
  "description": "근무 신청 4",
  "rejections": 0,
  "createdAt": "2024-11-12T07:38:56.839Z",
  "__v": 0
},
{
  "_id": "67330619c896927f8ab7dd20",
  "workerId": null,
  "name": "최지우",
  "start": "2024-11-12T09:00:00.000Z",
  "end": "2024-11-12T12:00:00.000Z",
  "lastShiftStart": "2024-10-28T09:00:00.000Z",
  "lastShiftEnd": "2024-10-28T12:00:00.000Z",
  "status": "Approved",
  "description": "근무 신청 5",
  "rejections": 0,
  "createdAt": "2024-11-12T07:39:05.548Z",
  "__v": 0
},
}
```

개발 현황 : Backend

알고리즘 개발

```
// 근무자 선발 알고리즘 함수
const selectWorkers = async () => {
  const requests = await ShiftRequest.find({ status: 'Pending' });
  const MAX_WORKERS = 3; // 근무자 최대 인원
  // 점수 계산 및 정렬
  const sortedRequests = requests
    .map(request => {
      // 마지막 근무로부터 경과한 시간 (시간 단위)
      const hoursSinceLastShift = (new Date() - new Date(request.lastShiftEnd)) / (1000 * 60 * 60);

      // 점수 계산: 마지막 근무 경과 시간 + 거절 횟수에 따른 가산점
      const score = hoursSinceLastShift + request.rejections * 5;
      return { ...request.toObject(), score };
    })
    .sort((a, b) => b.score - a.score); // 높은 점수 순으로 정렬

  // 선발된 근무자와 거절된 근무자 분리
  const selectedWorkers = sortedRequests.slice(0, MAX_WORKERS);
  const rejectedWorkers = sortedRequests.slice(MAX_WORKERS);

  // 근무자 상태 업데이트 You, 22 minutes ago • Uncommitted changes
  for (const worker of selectedWorkers) {
    await ShiftRequest.findByIdAndUpdate(worker._id, { status: 'Approved' });
  }
  for (const worker of rejectedWorkers) {
    await ShiftRequest.findByIdAndUpdate(worker._id, { status: 'Rejected' });
  }

  // 선발된 근무자와 거절된 근무자를 필요한 형태로 반환
  return {
    selectedWorkers: selectedWorkers.map(worker => ({
      name: worker.name,
      score: worker.score
    })),
    rejectedWorkers: rejectedWorkers.map(worker => ({
      name: worker.name,
      score: worker.score
    }))
  };
};
```

[근무인원 선발함수 (selectWorkers)]

Step 1) 최대 근무 인원수 설정 (ex. 3명)

Step 2) 동 시간대 근무 희망자들의 신청 거절된 횟수,
마지막 근무로부터 경과한 시간을 기반으로 점수 계산

Step 3) 점수 내림차순정렬, 최대 근무 인원수를
기준으로 자르고, 상태를 업데이트
(Approved이면 초기화, Rejected이면 +1)

Step 4) 선발or거절된 인원, 점수 return

1)지원자 점수 계산:

점수는 최근 일한 시간이고 지난 근무 횟수를 기준으로 하여, 오래 일하지 않은 사람에게 더 높은 점수를 부여
배정에서 여러 번 밀려난 지원자에게 추가 점수를 부여하여 다음 근무 배정시 우선순위와 배정 확률을 높임.

2)지원자 정렬:

특정 시간대에 대한 선호도가 높은경우(신청인원 > 예정인원) 높은 점수 순으로 근무자를 배정
동점자가 있을 경우 동점자 사이에서 무작위로 배정.

3)근무자 배정:

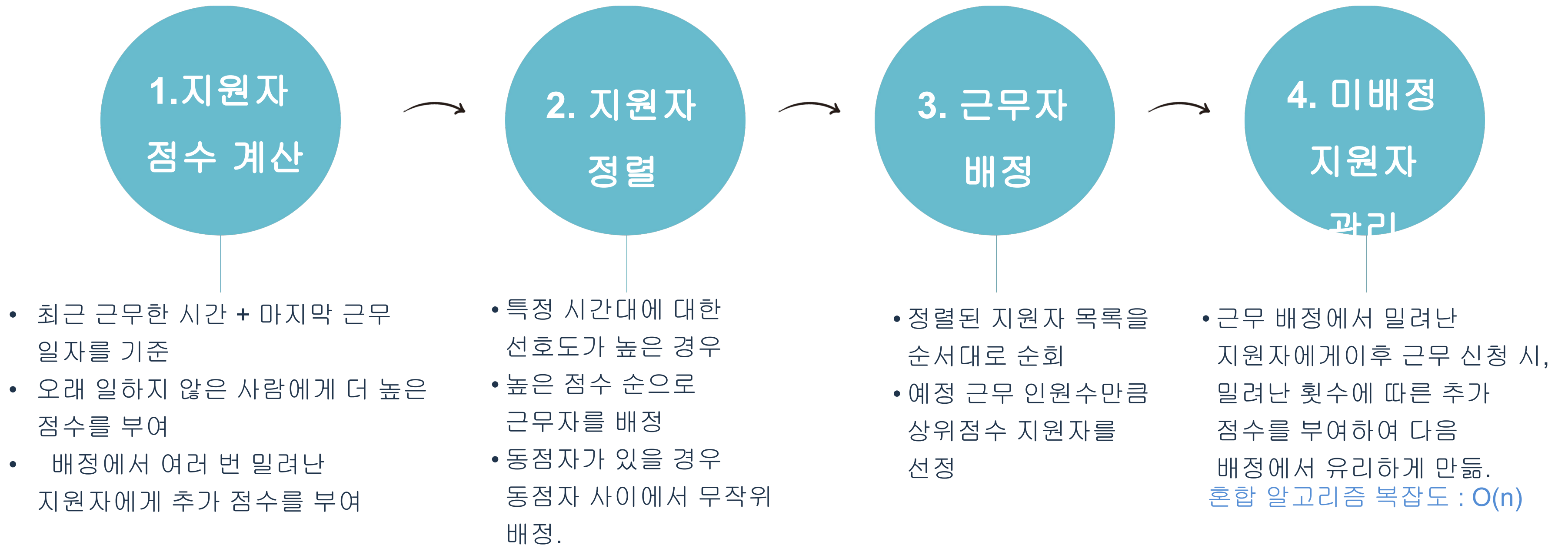
정렬된 지원자 목록을 순서대로 순회하여 예정 근무 인원수만큼 상위점수 지원자를 선정
점수와 마지막 근무 시간을 기준으로 배정된 사람이 결정되므로, 공정한 근무 배분이 가능

배정되지 않은 지원자 관리:

근무 배정에서 밀려난 지원자에게는 이후 근무 신청 시, 우선순위를 점유하기 위해 밀려난
횟수에 따른 추가 점수를 부여하여 다음 배정에서 유리하도록 합니다.

근무 일정 배정 알고리즘 구현 계획

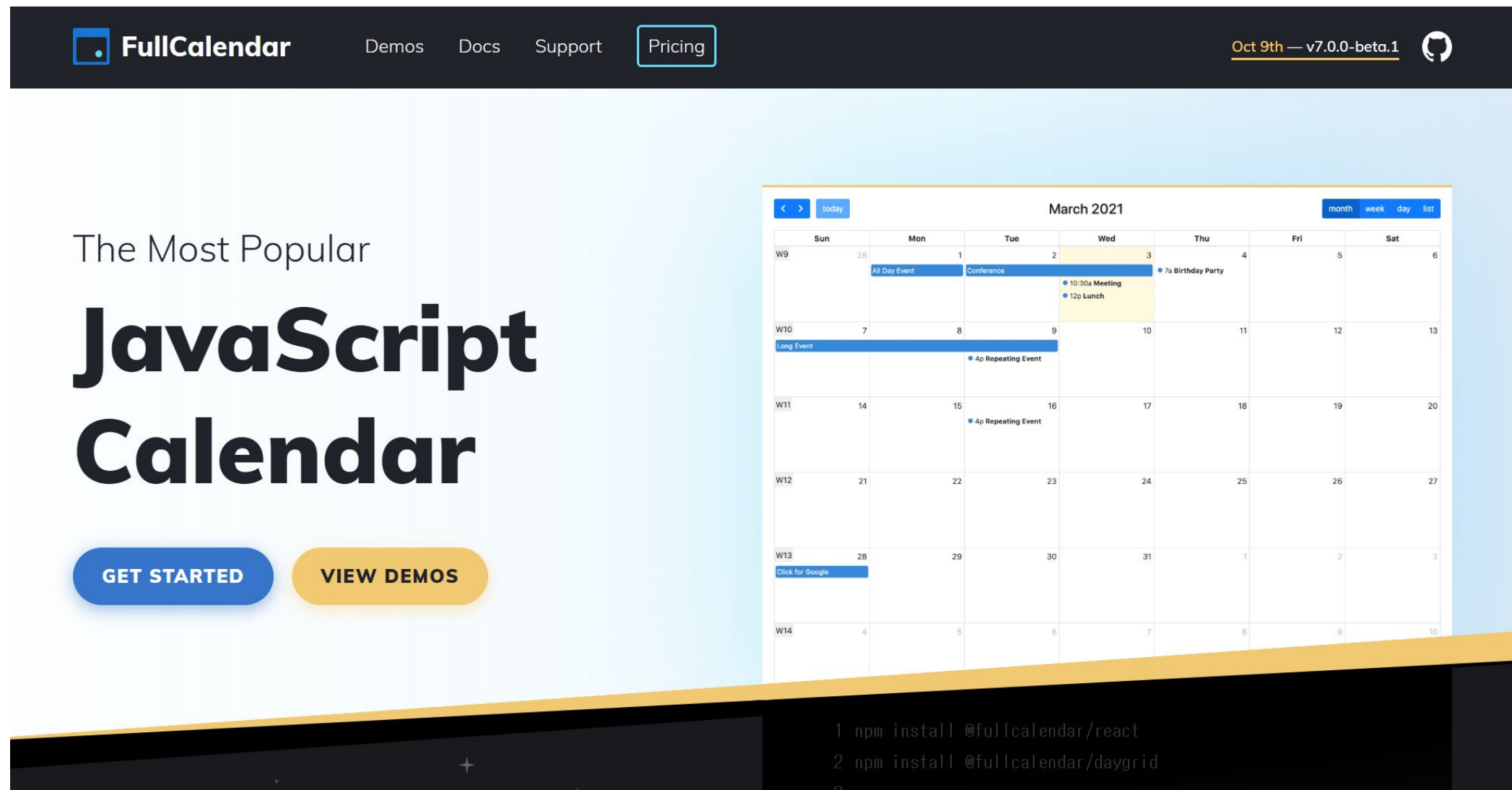
Fair queuing + Aging algorithm 혼합



2. 개발 현황

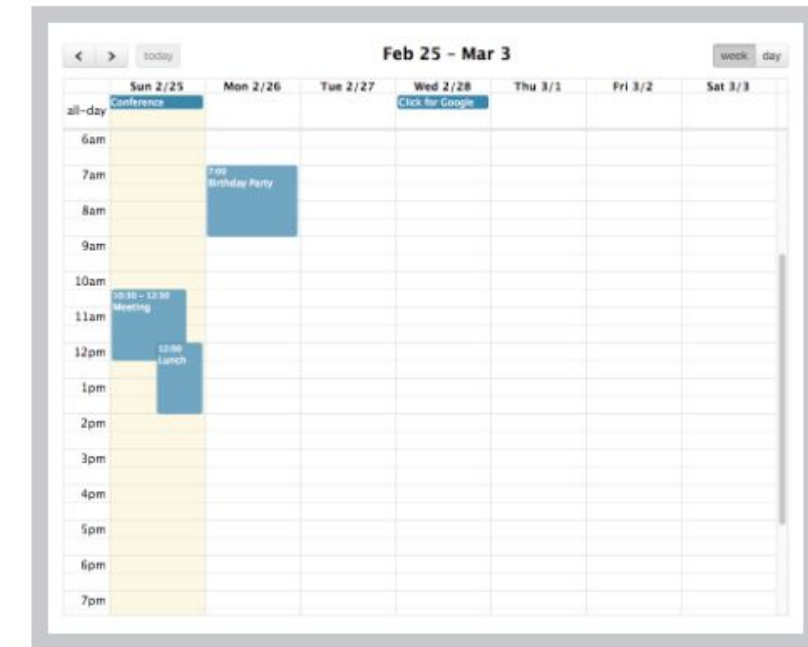
개발 현황 : FrontEnd

-오픈소스 라이브러리 FullCalendar



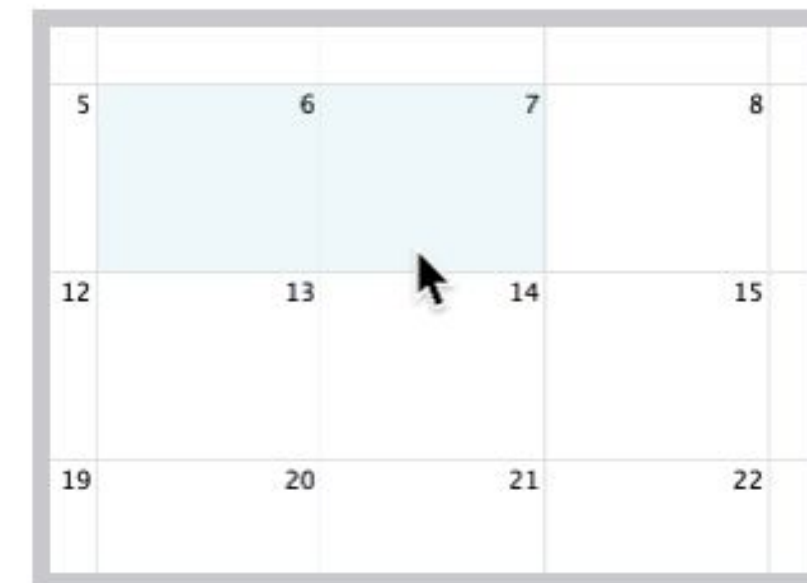
The screenshot shows the FullCalendar website. The header includes the FullCalendar logo, navigation links (Demos, Docs, Support, Pricing), and the current version (Oct 9th — v7.0.0-beta.1). The main content area features the text "The Most Popular JavaScript Calendar" and two buttons: "GET STARTED" and "VIEW DEMOS". A calendar widget is displayed, showing the month of March 2021. The calendar has a header with navigation arrows, a "today" button, and view toggles (month, week, day, list). The calendar grid shows dates from Sunday to Saturday. Events are shown as blue bars with labels: "All Day Event", "Conference", "10:30a Meeting", "12p Lunch", "7a Birthday Party", "Long Event", and "4p Repeating Event".

TimeGrid View



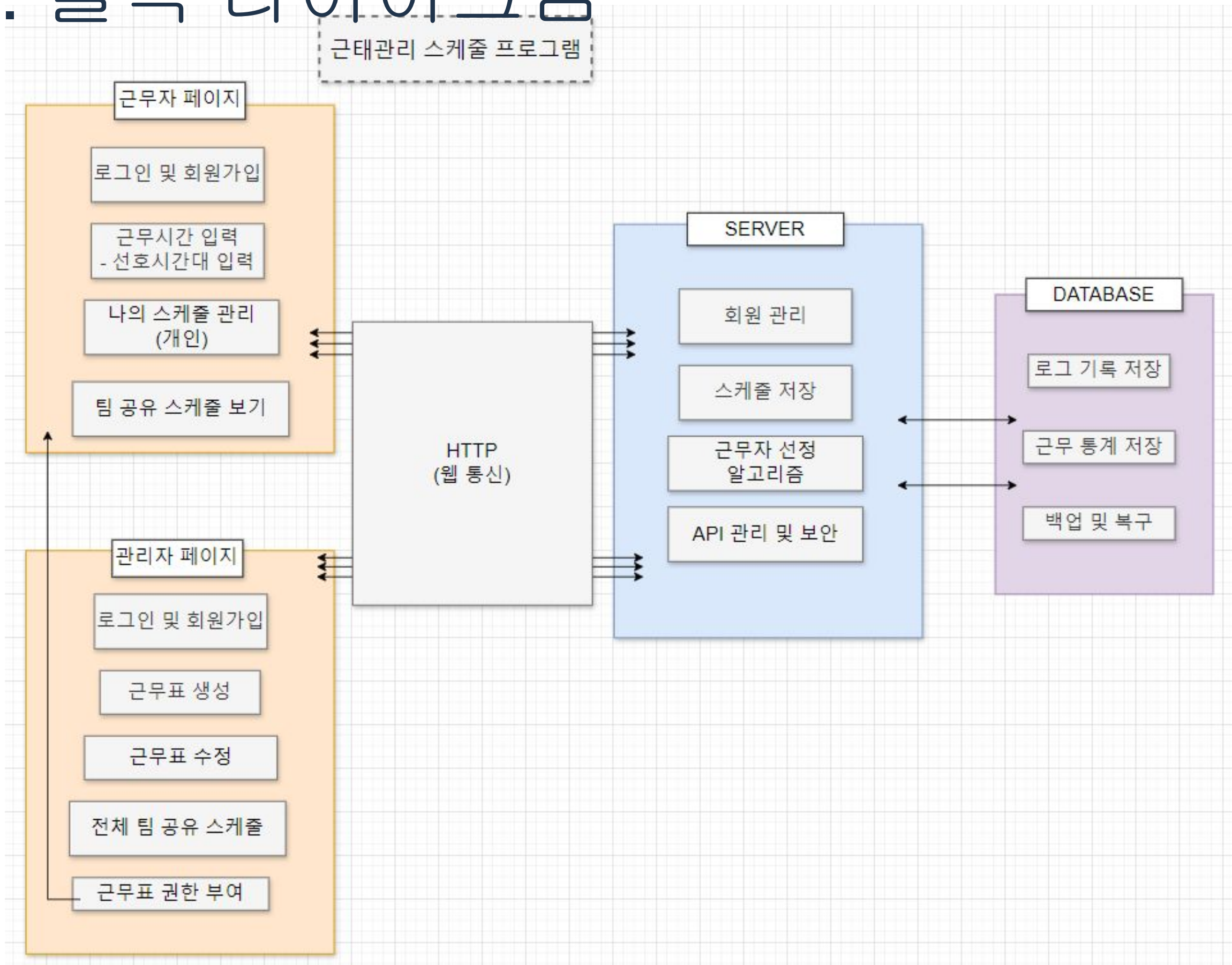
timeGridWeek and timeGridDay views

Date Clicking & Selecting



dateClick and selectable dates/times

부록3. 블록 다이어그램



근무자, 관리자 페이지로
나누어 구현

서비스 개발 배경 및 필요성

업무 효율성 측면

근무자들의 근무 가능 시간대를
취합해 수동으로 근무 일정을
짜기 위한 시간과 인력 소모로
업무 효율성이 감소

근무 관리 정확도 측면

관리자가 직접 근무 일정을 관리해
중복이나 누락 등의 오류 발생

근무 인원 측면

비용 절감을 위해 매장 근무자를
늘리거나 근무지 이동이 잦아짐



근무 만족도 측면

근무자들이 원하는 근무 시간대가
반영되지 않는 근무 일정으로 인해
근무 만족도 하락

근무 편성의 공정성

측면 근무편성이 선착순으로
정해지거나 특별한 기준이
없어 근무 일정 편성의 공정성
낮음

근무 편성으로 인한

갈등 근무 일정 편성과 변경 과정에서
관리자와 근무자, 근무자와 근무자
간의 갈등 발생

선행사례 조사

평가 항목	Shiftee (시프티)	보스몬	Pinpl(핀 플)	Dutying	HowMuch (하우머치)	ShiftMate
근무시간 확인	O	O	O	O	O	O
근무자-관리자 간 통합 일정공유	X	X	X	O	X	O
근무시간 선호도 신청 및 반영	X	△ (근무시간 변경 신청 및 승인기능만 존재)	X	O	X	O
지난 근무의 우선순위를 반영한 근무 배정	X	X	X	X	X	O

제약 사항 및 대안 도출



한정된 개발 기간

개발 기간이 이번 한기 내로 한정됨



전문성 부족

특정 직종의 근무 일정 관리에 대한
전문 지식 부족



알고리즘의 한계

모든 근무자의 선호 시간대를
반영할 수 없음



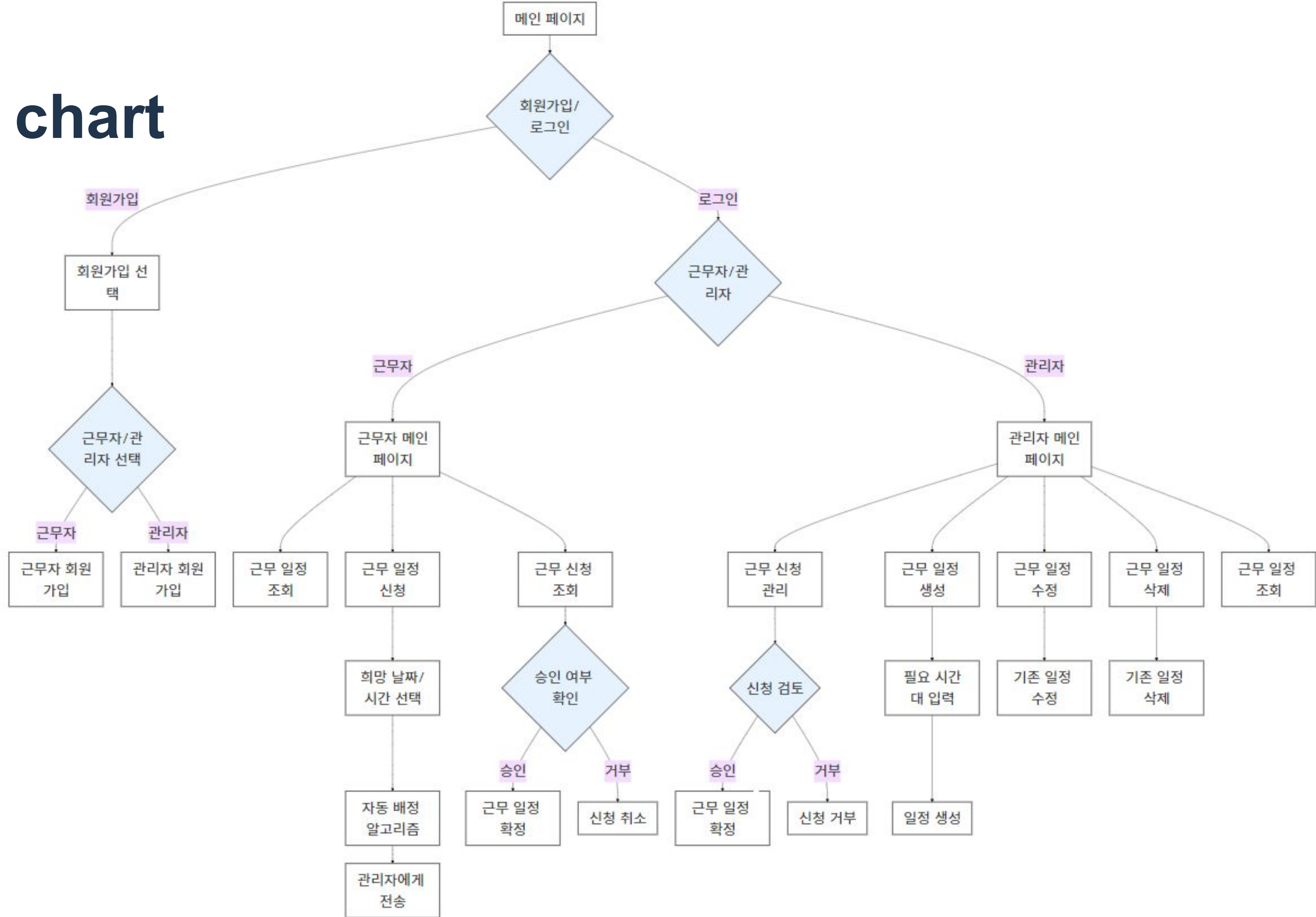
‘아르바이트’로 서비스 타겟 한정

- 불특정 다수의 스케줄 근무 아르바이트 생을
주 타겟으로 설정
- 근무 시간 자동 배정 기능을 우선적으로
구현
- 급여 관리 등 부가 기능은 추후 추가

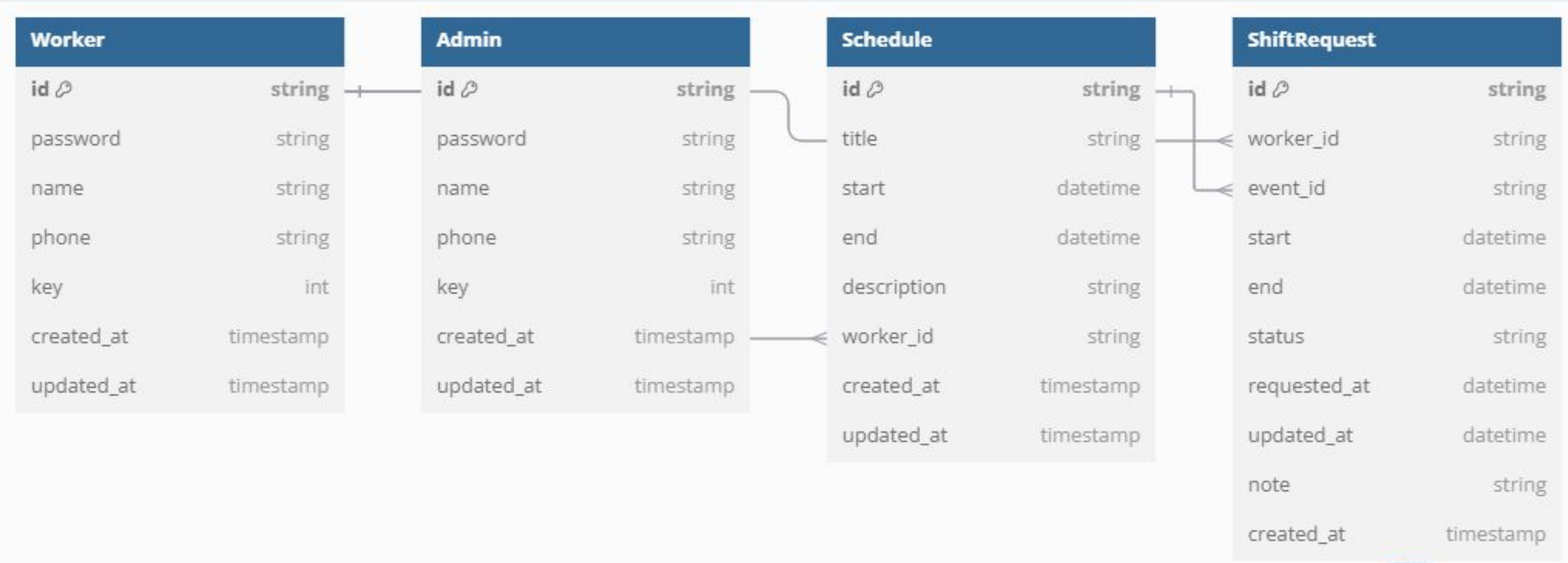
여러 번의 테스트 및 피드백 진행

- 최대한 다양한 조건으로 여러 번의 테스트
진행 계획
- 다수의 피드백을 통해 ‘최선’의 알고리즘
구현

Flow chart



E-R Diagram



API

API 명세서 ...

Aa Index	≡ 기능	≡ HTTP	≡ API path	+
<u>메인 페이지</u>	첫 접속 시 메인 화면	GET	/home	
<u>회원가입</u>	근무자/관리자 회원가입	POST	/home/signup	
<u>로그인</u>	근무자/관리자 로그인	POST	/home/login	
<u>근무자</u>	로그인 후 근무자 메인 페이지	POST	/worker/main	
	근무 일정 신청	POST	/worker/events/apply	
	근무 일정 신청 승인 조회	GET	/worker/events/apply/:workerId	
	근무 일정 조회	GET	/worker/events/all	
<u>관리자</u>	메인	POST	/admin/main	
	근무 일정 생성	GET POST	/admin/events/create	
	근무 일정 수정	GET	/admin/events/edit/:id	
	근무 일정 삭제	POST	/admin/events/delete/:id	
	근무 일정 조회	GET	/admin/events/all	
	근무 신청 승인	POST	/admin/events/approve/:requestId	
	근무 신청 거절	POST	/admin/events/reject/:requestId	

ShiftMate의 필요성 : 일정 배정의 어려움

ShiftMate개요 다이어그램 기능설명 시연 기대효과 및 확장성 추후 계획 QnA

"인건비 낮춰라"...인력 효율화 나선 스타벅스

인력·물류 효율화 방침 내걸어
바리스타 한 명이 여러 매장 근무
냉동 식자재 확충으로 배송비 절감
특화매장 가격 경쟁력 확충

입력 2024.10.13 18:33:32 수정 2024.10.13 18:33:32



프랜차이즈 카페에서 인력을 공유
근무자들의 일정 배정 및 변경 시 혼란 가중화 가능성

매일경제

뉴스 MK위클리연재 스타투데이 마켓 부동산 교육

三 뉴스홈 경제 기업 사회 국제 부동산 증권 정치 IT·과학 문화 오피니언 스포츠 기획·연재

경제

주휴수당 부담에 단기알바만 채용...편의점 한 곳에 6명 근무도

이희조 기자 love@mk.co.kr
김동은 기자 bridge@mk.co.kr
양연호 기자 yeonho8902@mk.co.kr

입력 : 2024-06-12 17:59:37 수정 : 2024-06-12 19:56:23



5월 단기 취업자 역대 최대
비용 줄이려는 자영업자들
가족 동원에 키오스크까지
'직원 없이 영업' 교육책도
최저임금 도입 취지와 달리
소상공인·근로자 피해 키워

매장당 근무자 수의 증가로 인해 일정 관리에 혼선
발생

주휴 수당으로 인해 근무자 당 근무시간이 줄고
근무자 수가 늘어남에 따라 일정 배정 및 변경에

유스케이스 다이어그램

