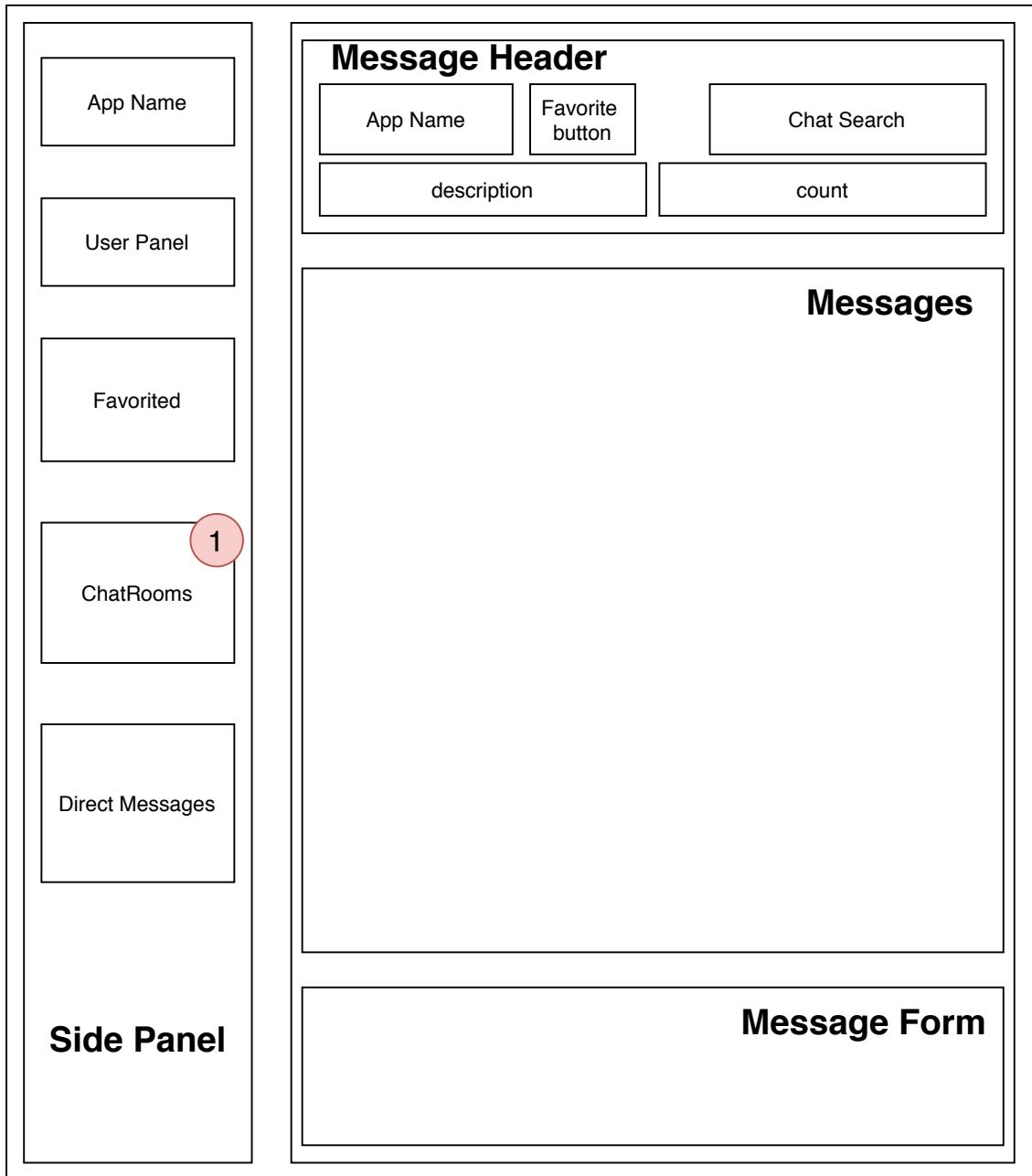


채팅 페이지 UI



1

채팅 룸 생성 Modal



JS	Favorited.js	U
JS	SidePanel.js	U
JS	UserPanel.js	U

```
function SidePanel(props) {
  return (
    <div style={{|
      backgroundColor: "#7B83EB",
      padding: '2rem',
      minHeight: '100vh',
      color: 'white',
      minWidth: '275px'      John Ahn, 11 days ago • Fav
    }}>
      <UserPanel currentUser={props.currentUser} />

      <Favorited currentUser={props.currentUser} />

      <ChatRooms />

      <DirectMessages currentUser={props.currentUser} />
    </div>
  )
}
```

MainPanel

✓	MainPanel	●
JS	MainPanel.js	U
JS	Message.js	U
JS	MessageForm.js	U
JS	MessageHeader.js	U

```
<div style={{ padding: '2rem 2rem 0 2rem' }}>
  <MessageHeader/>

  <div style={{|
    width: '100%',|
    height: '450px',|
    border: '.2rem solid #ececce'.|
  }}>
```

```
border: 1px solid #e0e0e0,
borderRadius: '4px',
padding: '1rem',
marginBottom: '1rem',
overflowY: 'auto'
}}>
</div>

<MessageForm />
</div>
```

UserPanel
Favorited
ChatRooms
DirectMessages

MessageHeader
Message
MessageForm

User Panel 기본 구조 만들기

Side Panel 기본 스타일링

Logo

```
/* Logo */
<h3 style={{ color: 'white' }}>
  <IoIosChatboxes />
  {" "} Chat App
</h3>
```

User Dropdown

```
/* User Dropdown */
<div style={{ display: 'flex', marginBottom: '1rem' }}>
  <Image
    style={{ width: '30px', height: '30px', marginTop: '3px' }}
    roundedCircle
  />
  <Dropdown>
    <Dropdown.Toggle
      style={{ background: 'transparent', border: '0px' }}
      id="dropdown-basic"
    >
      {user && user.displayName}
    </Dropdown.Toggle>

    <Dropdown.Menu>
      <Dropdown.Item onClick={handleOpenImageRef}>
        프로필 사진 변경
      </Dropdown.Item>
      <Dropdown.Item onClick={handleLogout}>
        로그아웃
      </Dropdown.Item>
    </Dropdown.Menu>
  </Dropdown>
</div>
```

유저 정보 가져오기

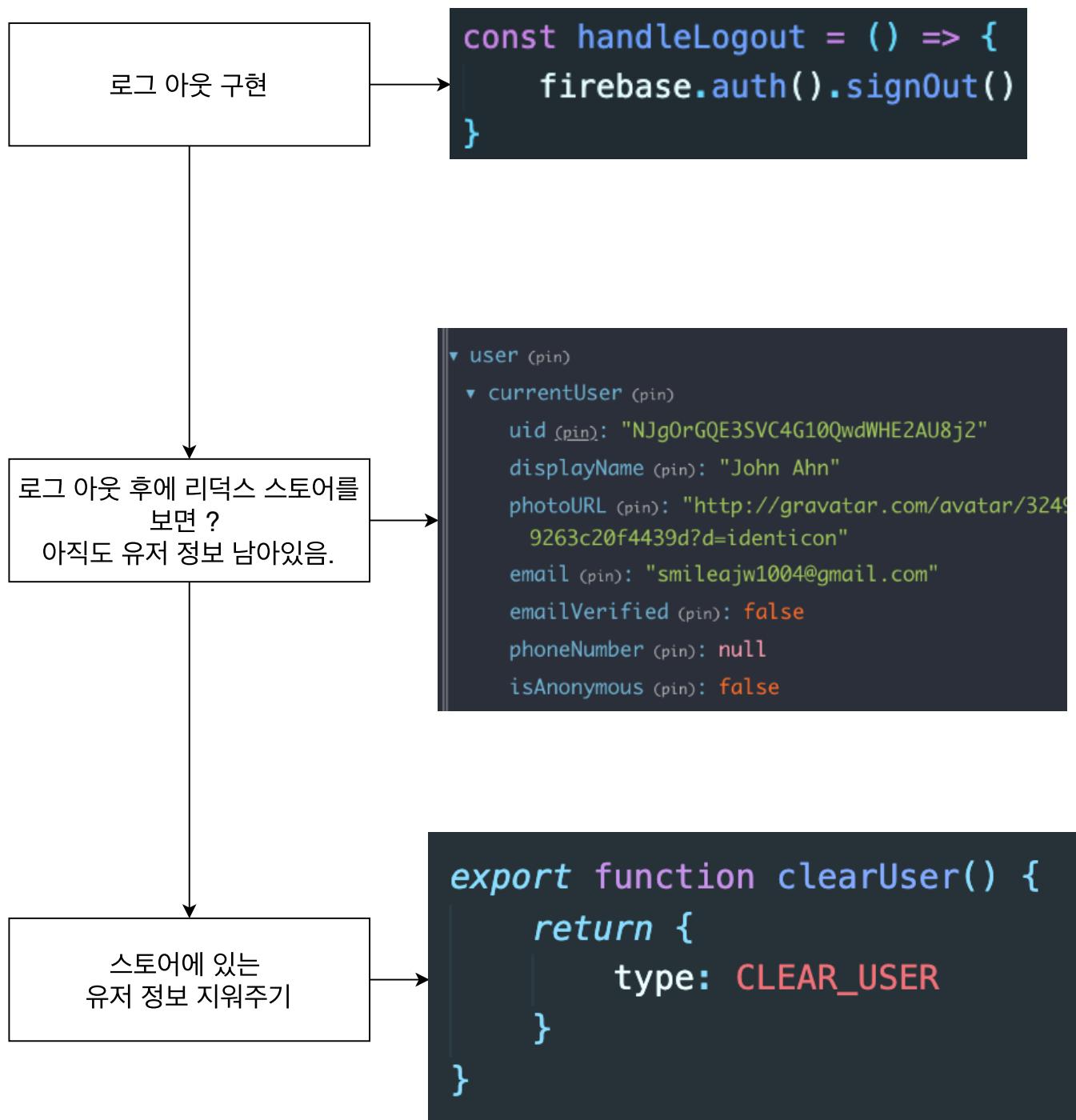
```
const user = useSelector(state => state.user.currentUser)
```



유저 정보 넣어주기

```
graph TD; A[유저 정보 넣어주기]
```

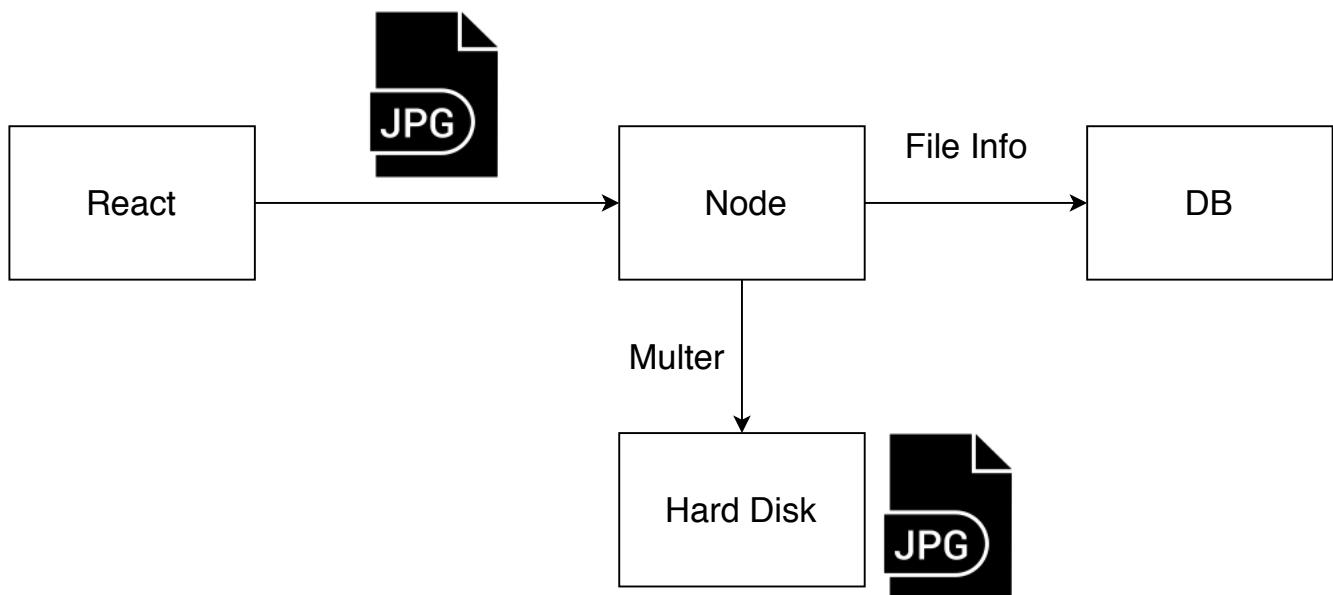
로그 아웃 & 리덕스 스토어에서 유저 정보 지워주기



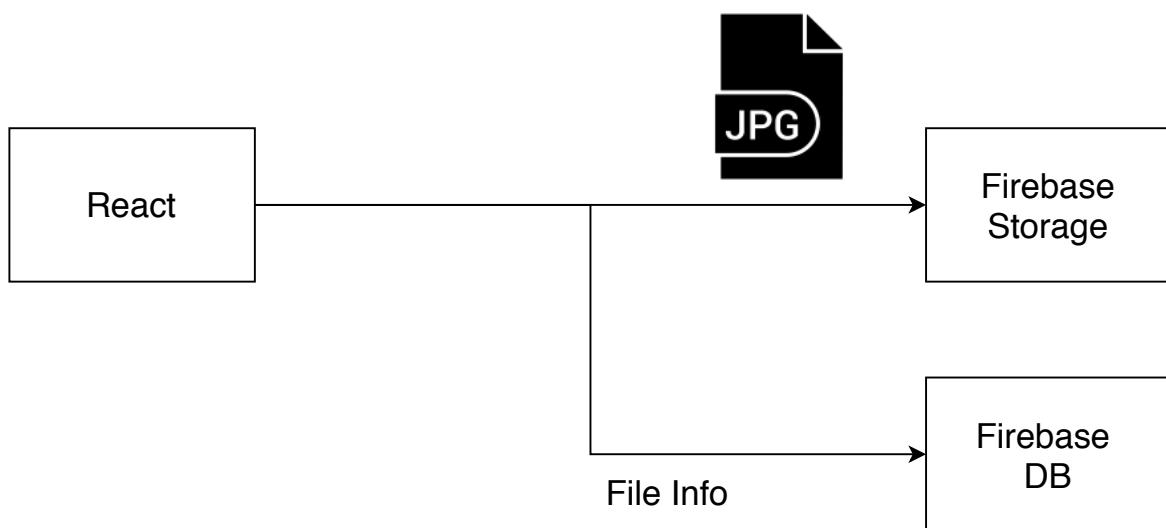
프로필 이미지 수정

유저 이미지를 교체하려면
먼저 생각해 봐야 할 것들...

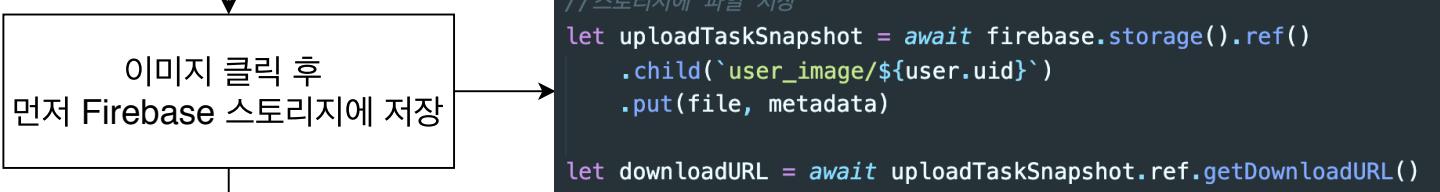
React + Node + DB



React + Firebase ?

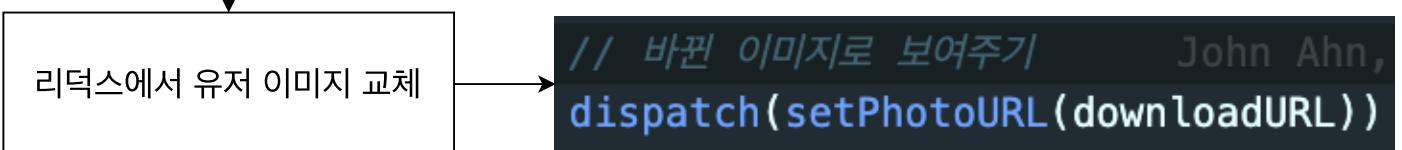
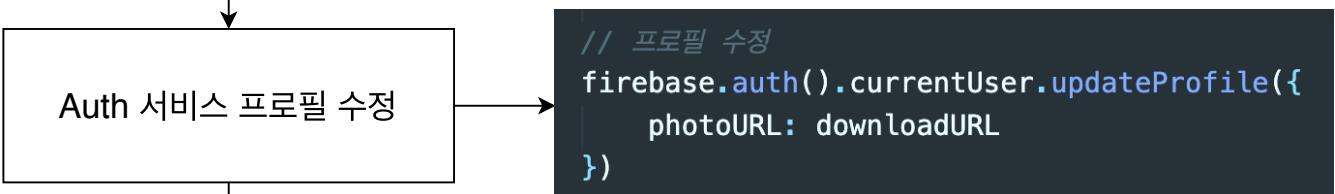






Screen Shot 2020-10-24 at 16.16.19.png

{*contentType*: "image/png"}



(method) `firebase.storage.Reference.put(data: Blob | Uint8Array | ArrayBuffer, metadata?: firebase.storage.UploadMetadata): firebase.storage.UploadTask`

Uploads data to this reference's location.

`@param data` — The data to upload.

`@param metadata`

Metadata for the newly uploaded object.

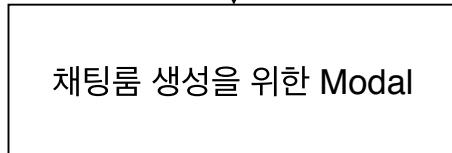
`@return`

An object that can be used to monitor and manage the upload.

채팅 룸 리스트 UI

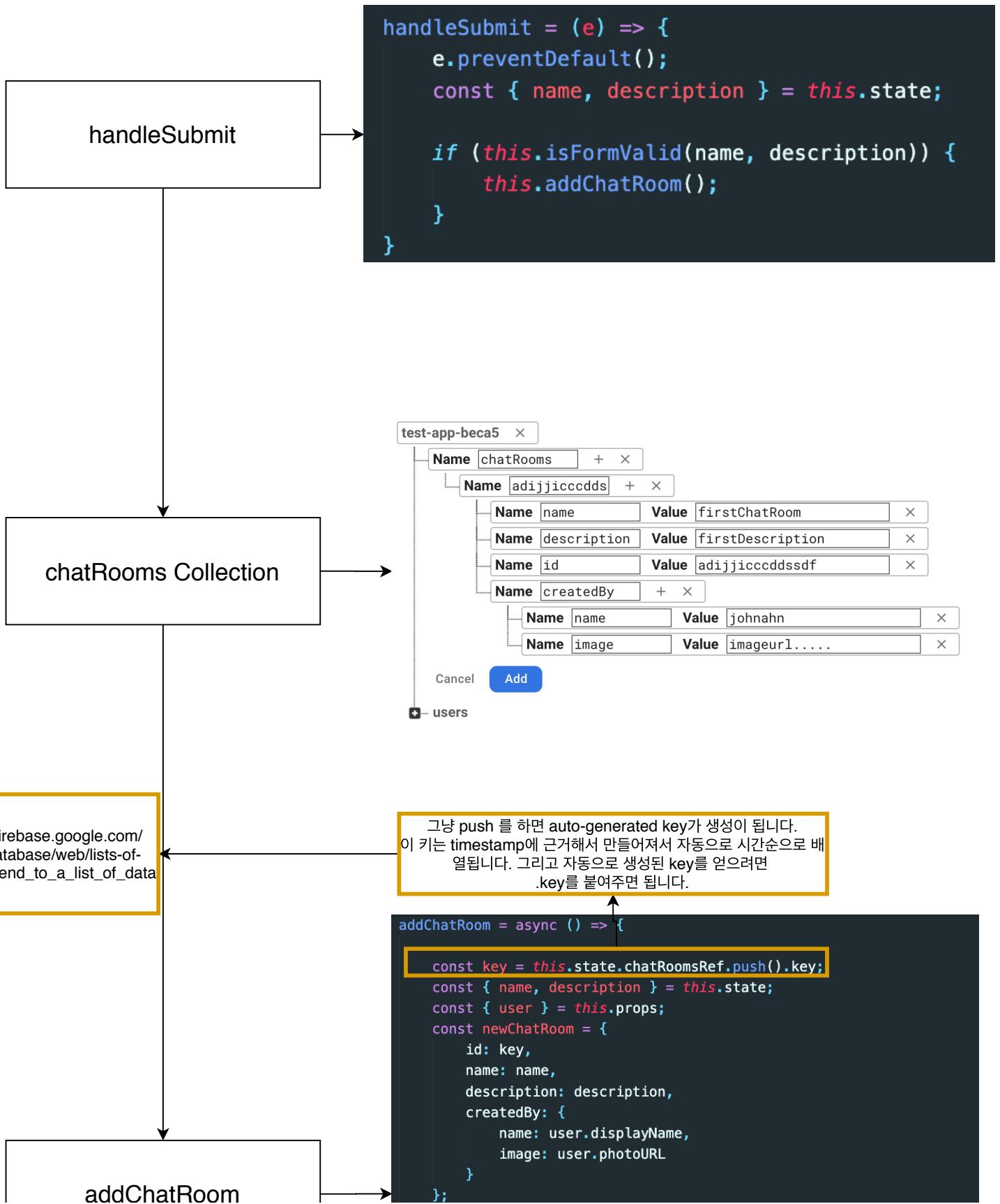


```
<div style={{  
    position: 'relative', width: '100%',  
    display: 'flex', alignItems: 'center'  
}}>  
    <FaRegSmileWink style={{ marginRight: 3 }} />  
    CHAT ROOMS{" " } ({chatRooms.length})  
  
    <FaPlus  
        style={{  
            position: 'absolute',  
            right: 0, cursor: 'pointer'  
        }}  
        onClick={this.handleShow}  
    />  
</div>
```



```
{/* ADD ChatRoom Modal */}  
<Modal show={show} onHide={handleClose}>  
    <Modal.Header closeButton>  
        <Modal.Title>Create a chat room</Modal.Title>  
    </Modal.Header>  
    <Modal.Body>  
        <Form onSubmit={handleSubmit}>  
            <Form.Group controlId="formBasicEmail">  
                <Form.Label>Name</Form.Label>  
                <Form.Control  
                    onChange={(e) => setName(e.target.value)}  
                    type="text"  
                    placeholder="Enter chat room name"  
                />  
            </Form.Group>  
  
            <Form.Group controlId="formBasicPassword">  
                <Form.Label>Description</Form.Label>  
                <Form.Control  
                    onChange={(e) => setDescription(e.target.value)}  
                    type="text"  
                    placeholder="Enter chat room description"  
                />  
            </Form.Group>  
        </Form>  
    </Modal.Body>  
    <Modal.Footer>  
        <Button variant="secondary" onClick={handleClose}>  
            Close  
        </Button>  
        <Button variant="primary" onClick={handleSubmit}>  
            Save Changes  
        </Button>  
    </Modal.Footer>  
</Modal>
```

ChatRoom 생성하기



```
try {
    await this.state.chatRoomsRef.child(key).update(newChatRoom);
    this.setState({
        name: '',
        description: '',
        show: false
    })
} catch (error) {
    alert(error)
}

};
```

Firebase에서 데이터 실시간으로 받기

firebase 데이터베이스에서
데이터를 가져오는 방법

저장된 데이터를 listener로
실시간 데이터를 받을수있습니다.

Event Listener

Add된 데이터를
Listener로 받기

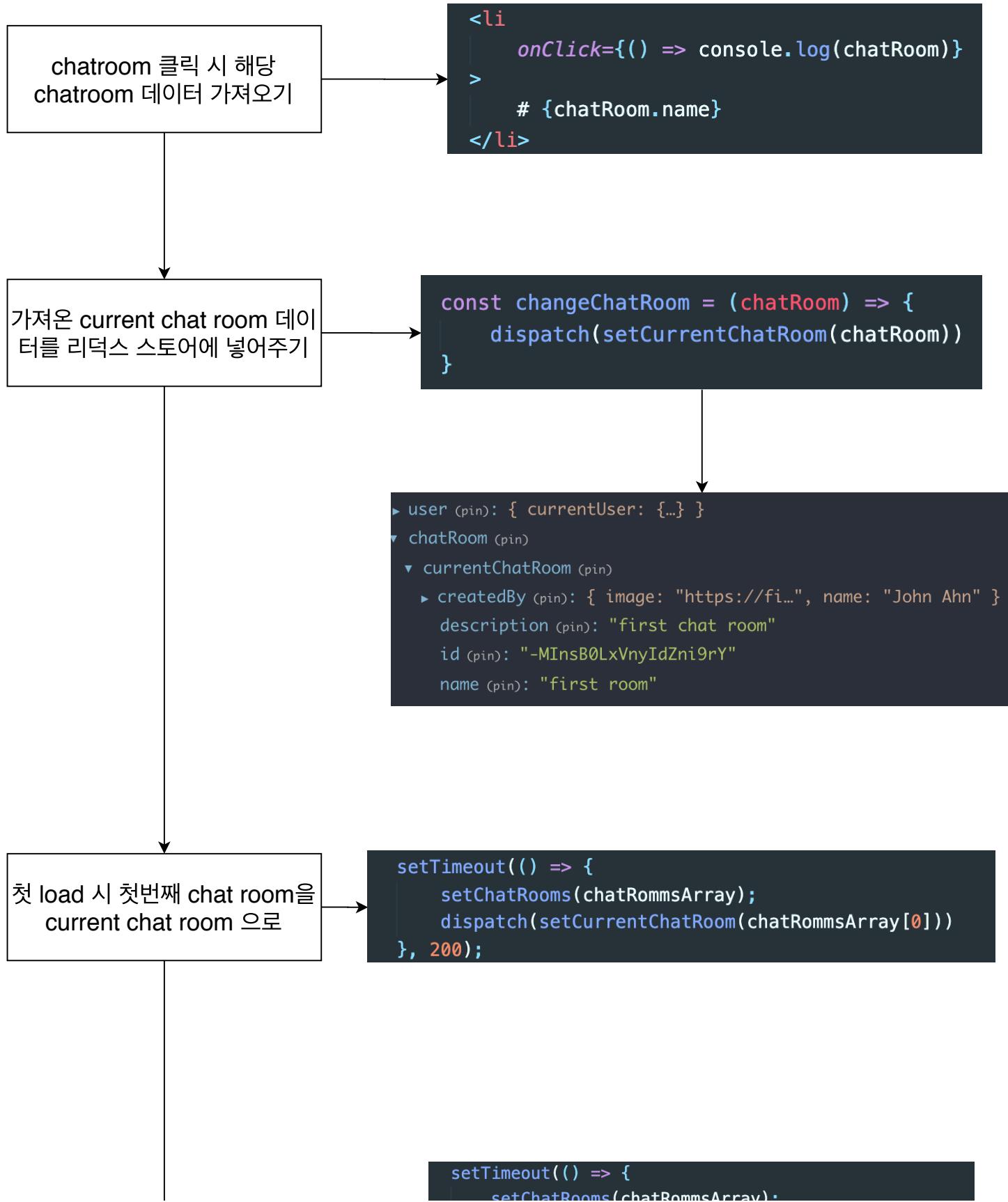
```
AddChatRoomsListeners = () => {
  let chatRoomsArray = []
  // chatRoomsRef.on(eventType, callback)
  this.state.chatRoomsRef.on("child_added", DataSnapshot => {
    chatRoomsArray.push(DataSnapshot.val());
    this.setState({ chatRooms: chatRoomsArray });
  })
}
```

chatRooms 데이터
화면에 보여주기

```
<ul style={{ listStyleType: 'none', padding: '0' }}>  
  {renderChatrooms(chatRooms)}  
</ul>
```

```
const renderChatrooms = chatRooms =>  
  chatRooms.length > 0 &&  
  chatRooms.map(chatRoom => (  
    <li  
      onClick={() => console.log(chatRoom)}  
    >  
      # {chatRoom.name}  
    </li>  
  ));
```

Set current chat room



active chat room은
다른 색상 주기

```
setChatRooms(chatRoomsArray),
// 첫번째 chatRoom을 current chat room으로
dispatch(setCurrentChatRoom(chatRoomsArray[0]))
setActiveChatRoomId(chatRoomsArray[0].id)
}, 200);
```

```
style={{
  backgroundColor: room.id === activeChatRoomId && "#fffff45"
}}
```

```
const changeChatRoom = (chatRoom) => {
  dispatch(setCurrentChatRoom(chatRoom))
  setActiveChatRoomId(chatRoom.id)
}
```

clean up event listener

