나의 앱 만들기-

**완료보고서**

|  |  |
| --- | --- |
| **과목** | **모바일프로그래밍** |
| **담당 교수** | **강운구** |
| **제출일자** | **2020.06.26** |
| **학과** | **컴퓨터공학과** |
| **학번** | **201835756** |
| **이름** | **최희연** |

**<목차>**

**0. 애플리케이션 개요---------------------------------------p. 3**

**1. 애플리케이션 구성---------------------------------------p. 4-5**

**2. 애플리케이션 실행화면-----------------------------------p. 6-7**

**3. 애플리케이션 구현----------------------------------------p. 8-23**

**4. 후기--------------------------------------------------------p. 24**

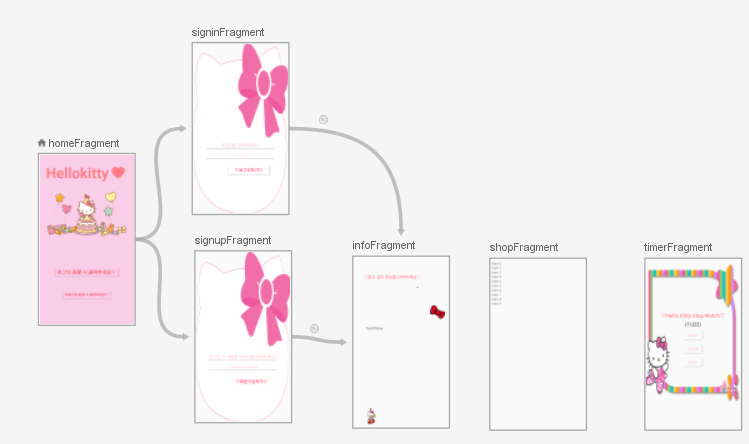
**0. 애플리케이션 개요**

**1) 이름: HellokittyInformation**

**2) 개요**

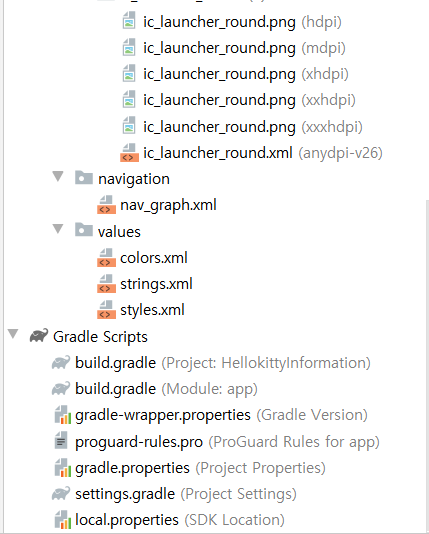
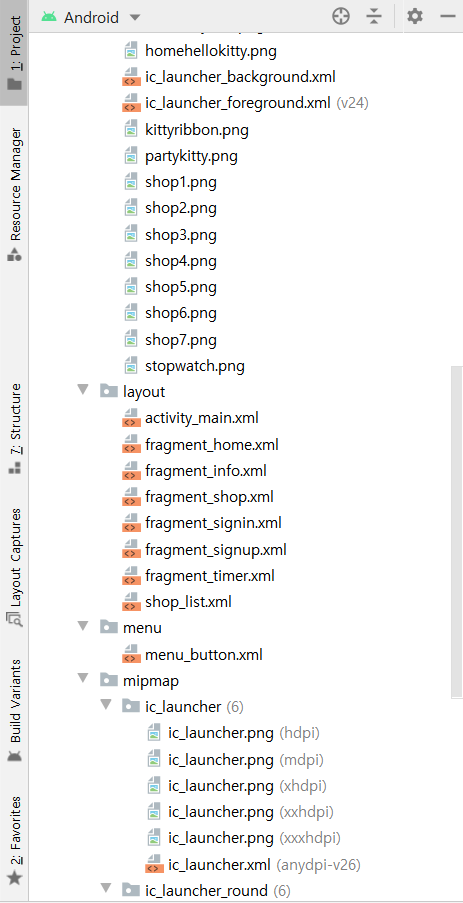
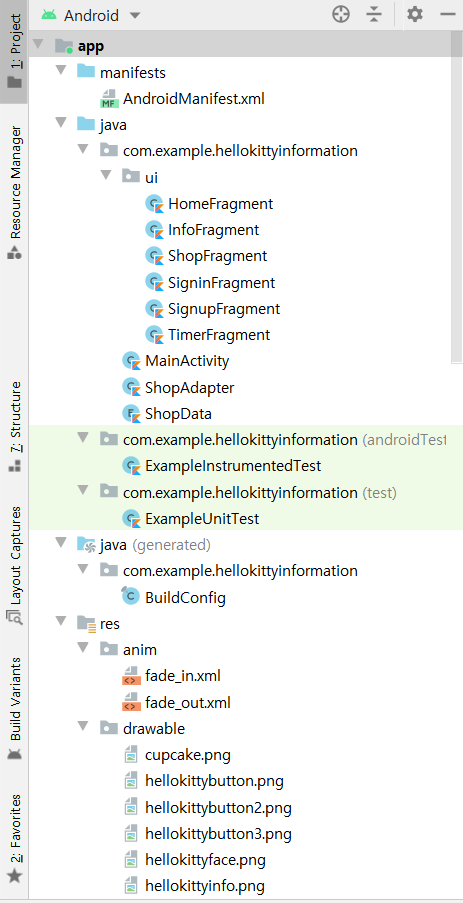
헬로키티를 정말 사랑하는 본인은 플레이 스토어에 관련 앱이 있나 검색해보았는데, 몇몇 게임말고는 없었다. 헬로키티를 정말 사랑하는 사람들을 위한 앱을 개발하고 싶다고 생각하게 되었고 헬로키티 관련 정보를 담고, 관련 쇼핑몰을 알려주며, 기분 좋게 헬로키티 배경화면을 가진 타이머를 개발하고자 하였다.

**\*주요기능**

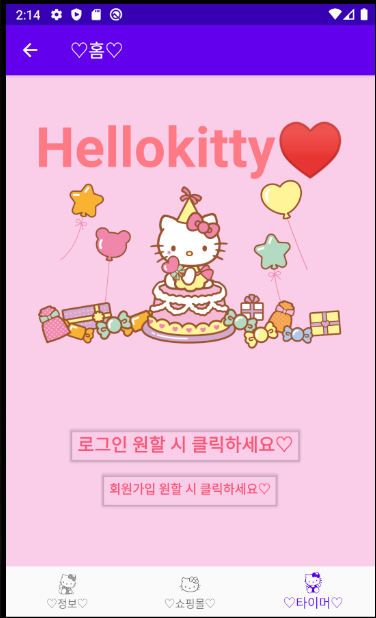


**1. 애플리케이션 구성**

**1) 프로젝트 구성도**



**2) 메인 화면 구성도**

로그인이나 회원가입을 할 수 있는 화면을 두었다. 건너뛰고 그냥 앱을 실행해도 된다.

앱을 실행하려면 아래에 메뉴바에서 세 가지 메뉴 중에 선택하여 이용하면 된다.

**3) 실행가능**

|  |  |
| --- | --- |
| **주요 기능** | **상세 기능** |
| 로그인 | 로그인할 수 있음 |
| 회원가입 | 회원가입 할 수 있음 |
| 정보 | 헬로키티 관련 정보에 대해 볼 수 있음 |
| 쇼핑몰 | 쇼핑몰 목록과 클릭시 이동 가능한 링크 |
| 타이머 | 헬로키티 배경의 타이머 |

**2. 애플리케이션 실행 화면**

|  |  |
| --- | --- |
| **화면1** | **회면2** |
|  |  |
| **메인화면** | **로그인** |
| **화면3** | **화면4** |
|  |  |
| **회원가입** | **정보** |
| **화면5** | **회면6** |
|  |  |
| **쇼핑몰** | **쇼핑몰 링크 클릭 시** |
| **화면7** |  |
|  |  |
| **타이머** |  |

**3. 애플리케이션 구현**

**1) 메인화면**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | 메인화면에서는 제목과 이미지 그리고 로그인 선택 시 로그인을 할 수 있도록,  회원가입 선택시 회원가입을 할 수 있도록 하였다. |
| **HomeFragment.kt** | |
| **package** com.example.hellokittyinformation.ui  **import** android.os.Bundle **import** androidx.fragment.app.Fragment **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** androidx.navigation.fragment.findNavController **import** com.example.hellokittyinformation.R **import** kotlinx.android.synthetic.main.fragment\_home.view.\*  **class** HomeFragment : Fragment() {   *//HomeFragment에 출력할 뷰를 생성하여 반환* **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  *// Inflate the layout for this fragment* **val** rootView = inflater.inflate(R.layout.*fragment\_home*, container, **false**)   */\* 로그인 버튼을 눌렀을 때 이벤트 처리 \*/* rootView.btn\_signin.setOnClickListener **{** */\* NavController를 이용하여 navigate() 인자로 전달된 action의 destination으로 이동  - homeFragment에서 signinFragment로 이동  \*/  findNavController*().navigate(R.id.*action\_homeFragment\_to\_signinFragment*)  **}** */\* 회원가입 버튼을 눌렀을 때 이벤트 처리 \*/* rootView.btn\_signup.setOnClickListener **{** *//homeFragment에서 signupFragment 화면으로 이동  findNavController*().navigate(R.id.*action\_homeFragment\_to\_signupFragment*)  **}  return** rootView  } }*//end of HomeFragment* | |

|  |
| --- |
| **MainActivity.kt** |
| **package** com.example.hellokittyinformation  **import** androidx.appcompat.app.AppCompatActivity **import** android.os.Bundle **import** androidx.navigation.findNavController **import** androidx.navigation.ui.AppBarConfiguration **import** androidx.navigation.ui.NavigationUI **import** com.example.hellokittyinformation.R **import** kotlinx.android.synthetic.main.activity\_main.\*  **class** MainActivity : AppCompatActivity() {   **override fun** onCreate(savedInstanceState: Bundle?) {  **super**.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   NavigationUI.setupWithNavController(  bottom\_navigation, *findNavController*(R.id.*navigation\_host*)  )   NavigationUI.setupActionBarWithNavController(  **this**,  *findNavController*(R.id.*navigation\_host*),  *AppBarConfiguration*(  *setOf*(  R.id.*infoFragment*,  R.id.*shopFragment*,  R.id.*timerFragment* )  )  )  }*//end of onCreate* **override fun** onSupportNavigateUp() = *findNavController*(R.id.*navigation\_host*).navigateUp() } |

**2) 로그인**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | 아이디와 비밀번호를 입력하고 로그인 하기 버튼을 누르면 앱으로 입장할 수 있도록 하였다. |
| **SigninFragment.kt** | |
| **package** com.example.hellokittyinformation.ui  **import** android.os.Bundle **import** androidx.fragment.app.Fragment **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** androidx.navigation.fragment.findNavController **import** com.example.hellokittyinformation.R **import** kotlinx.android.synthetic.main.fragment\_signin.view.\*  **class** SigninFragment : Fragment() {   *//SigninFragment에 출력할 뷰를 생성하여 반환* **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  *// Inflate the layout for this fragment* **val** rootView = inflater.inflate(R.layout.*fragment\_signin*, container, **false**)   *//fragment\_signin\_btn 버튼 클릭 이벤트 처리* rootView.fragment\_signin\_btn.setOnClickListener **{** *//NavController를 이용하여 navigate() 인자로 전달된 action의 destination으로 이동  findNavController*().navigate(R.id.*action\_signinFragment\_to\_infoFragment*)  **}  return** rootView  } } | |

**3) 회원가입**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | 회원가입 시 사용할 아이디와 비밀번호 입력 후 회원가입하기 버튼 클릭시 앱으로 입장할 수 있도록 하였다. |
| **SignupFragment.kt** | |
| **package** com.example.hellokittyinformation.ui  **import** android.os.Bundle **import** androidx.fragment.app.Fragment **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** androidx.navigation.fragment.findNavController **import** com.example.hellokittyinformation.R **import** kotlinx.android.synthetic.main.fragment\_signup.view.\*  **class** SignupFragment : Fragment() {   *//SignupFragment에 출력할 뷰를 생성하여 반환* **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  *// Inflate the layout for this fragment* **val** rootView = inflater.inflate(R.layout.*fragment\_signup*, container, **false**)   *//fragment\_signup\_btn 버튼 클릭 이벤트 처리* rootView.fragment\_signup\_btn.setOnClickListener **{** *//NavController를 이용하여 navigate() 인자로 전달된 action의 destination으로 이동  findNavController*().navigate(R.id.*action\_signupFragment\_to\_infoFragment*)  **}  return** rootView  } } | |

**4) 정보**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | 스피너를 통해 정보부분을 구현하였다.  “♡이름♡” 부분을 눌러 다른 파트("♡이름♡", "♡역사♡", "♡키티의 가족♡", "♡키티의 친구♡","♡관련 사업♡")를 클릭하면 해당 정보를 알려준다. |
| **InfoFragment.kt** | |
| **import** android.os.Bundle **import** androidx.fragment.app.Fragment **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** android.widget.AdapterView **import** android.widget.ArrayAdapter **import** com.example.hellokittyinformation.R **import** kotlinx.android.synthetic.main.fragment\_info.\*  */\*\*  \* A simple* ***[Fragment]*** *subclass.  \*/* **class** InfoFragment : Fragment() {    **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {   *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.*fragment\_info*, container, **false**)  }   **override fun** onActivityCreated(savedInstanceState: Bundle?) {  **super**.onActivityCreated(savedInstanceState)   *//spinner에 표시할 아이탬* **var** list\_of\_items = *arrayOf*(**"♡이름♡"**, **"♡역사♡"**, **"♡키티의 가족♡"**, **"♡키티의 친구♡"**,**"♡관련 사업♡"**)   *//spinner에서 아이템을 선택하였을때 보여줄 summary* **var** list\_of\_summary = *arrayOf*(**"헬로키티: 일본의 캐릭터 회사 산리오에서 1974년에 출시한 오리지널 캐릭터. 일본의 대표 캐릭터이자 대중문화의 상징 중 하나로, 현재까지도 전세계 사람들에게 많은 사랑을 받고 있는 캐릭터이다. 전세계 미디어믹스 총 수입 2위(860억 달러)의 포켓몬스터 다음가는 괴물콘텐츠로 전 세계 미디어 믹스 프랜차이즈 규모, 현재 헬로키티의 판권 추정치(총 자산 가치)는 무려 20조가 넘으며, 각종 굿즈 및 엔터테인먼트 사업, 그리고 콜라보레이션 등의 라이센스 사업으로 해마다 약 4조원 이상을 벌고 있다."**,  **"최초로 키티를 탄생시킨 사람은 산리오의 디자이너였던 시미즈 유코(清水侑子)키티의 생모였지만 키티를 만든지 얼마 되지 않아서 산리오를 떠나게 되었다. 이후 2대 요네쿠보 세츠코(米窪節子)키티의 유모를 거쳐 3대 디자이너인 야마구치 유코(山口裕子)키운 엄마에 이르러 지금의 이미지와 위상이 정립되었다. 미디어에서는 사실상 야마구치 유코를 키티의 어머니로 부른다. 낳은 정 보다는 기른 정이 더 크다"**,  **"미미 화이트,조지 화이트,메리 화이트,앤서니 화이트,마거릿 화이트,타이니 참,챠미 키티,슈가"**,  **"디어 다니엘, 캐시, 몰리, 주디, 조이, 로티,로리, 팀&타미, 피피, 티피, 트레이시, 조디, 토마스"**,  **"무기[9], 약물[10][11], 담배[12] 빼고는 모든 것이 키티 제품으로 나왔다.고 할 정도로 상상을 초월하는 규모의 캐릭터 상품 시장이 형성되어 있다. 키티 와인이 나오면서 키티 술[13]까지 나왔다. 사람이 쓸 수 있는 물건의 거의 대다수를 차지하고 있는 것이다. 마음만 먹으면 헬로키티 제품으로 일상 생활을 도배하는 게 가능하며, 이렇게 다양한 상품 시장 덕에 키티 물건 몇 개만 사도 매니아가 될 수 있다."**)    *//어답터 설정 - 안드로이드에서 제공하는 어답터를 연결* spinner.*adapter* =  ArrayAdapter<String>(requireContext(), android.R.layout.*simple\_spinner\_dropdown\_item*, list\_of\_items)   *//아이템 선택 리스너* spinner.*onItemSelectedListener* = **object** : AdapterView.OnItemSelectedListener {   **override fun** onNothingSelected(p0: AdapterView<\*>?) {  *println*(**"알고싶은 정보를 클릭하세요."**)  }   *//아이템을 선택하였을 때* **override fun** onItemSelected(p0: AdapterView<\*>?, p1: View?, position: Int, p3: Long) {  *println*(**"당신이 알고 싶은 정보는 "** + list\_of\_items[position] + **"입니다."**)  *//선택한 정보에 대해 알려준다.* tv\_summary.*text* = list\_of\_summary[position]  }  }   } } | |

**5) 쇼핑몰**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | 리사이클러뷰를 통해서 쇼핑몰 목록을 구현하였다. 해당 쇼핑몰을 클릭하면 해당 쇼핑몰로 실제로 이동하는 것을 구현하고자 노력하였다.  최종으로 찾은 방법은 UI에 링크를 두어 클릭시 해당 쇼핑몰로 이동하는 것이었다. |
| **ShopFragment.kt** | |
| **package** com.example.hellokittyinformation.ui  **import** android.os.Bundle **import** androidx.fragment.app.Fragment **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** androidx.recyclerview.widget.LinearLayoutManager **import** androidx.recyclerview.widget.RecyclerView **import** com.example.hellokittyinformation.R **import** com.example.hellokittyinformation.ShopAdapter **import** com.example.hellokittyinformation.ShopData   */\*\*  \* A simple* ***[Fragment]*** *subclass.  \*/* **class** ShopFragment : Fragment() {   *//ShopData를 불러와 shop\_DataArray에 저장* **val shop\_DataArray**: ArrayList<ShopData> = ArrayList()  **lateinit var recyclerView**: RecyclerView     **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?   ): View? {  *// Inflate the layout for this fragment* **var** rootView= inflater.inflate(R.layout.*fragment\_shop*, container, **false**)    **recyclerView** = rootView.findViewById(R.id.*rv\_shop*!!)**as** RecyclerView  **recyclerView**.*layoutManager* = LinearLayoutManager(requireContext())  **recyclerView**.*adapter* = ShopAdapter(requireContext(),**shop\_DataArray**)   **return** rootView  }   override **fun** onViewCreated(view: View, savedInstanceState: Bundle?) {  **super**.onViewCreated(view, savedInstanceState)   } *//end of onViewCreated* }*//end Of ShopFragment* | |

|  |
| --- |
| **ShopData.kt** |
| **package** com.example.hellokittyinformation  **enum class** ShopData(**val imgRes**: Int, **val shop**:String, **val shopDetail**:String, **val shopUrl**:String){  **SHOP1**(R.drawable.*shop1*,**"♡홍이키티♡"**, **"3만원이상 구매시, 무료배송☆"**,**"http://hong2kitty.com/index.html"**),  **SHOP2**(R.drawable.*shop2*, **"♡buykitty♡"**,**"70~40% 세일 중 입니다☆"**,**"http://www.buykitty.co.kr/"**),  **SHOP3**(R.drawable.*shop3*, **"♡글루미캐릭터♡"**,**"캐릭터종합쇼핑몰 1위☆"**,**"http://www.gloomy.co.kr/"**),  **SHOP4**(R.drawable.*shop4*, **"♡재팬포스트♡"**,**"일본/미국 경매,구매대행 전문☆"**,**"https://japanpost.co.kr/"**),  **SHOP5**(R.drawable.*shop5*, **"♡키티체중계♡"**,**"헬로키티 체중계만을 취급하는 사이트☆"**,**"https://smartstore.naver.com/gifcall"**),  **SHOP6**(R.drawable.*shop6*, **"♡키티프렌즈♡"**,**"헬로키티 화장지만을 취급하는 사이트☆"**,**"https://smartstore.naver.com/hellokittyfriends"**),  **SHOP7**(R.drawable.*shop7*, **"♡키티마켓♡"**,**"귀욤♥귀욤♥ 정품 헬로키티 전문몰☆"**,**"https://smartstore.naver.com/bungae82"**)  } |

|  |
| --- |
| **ShopAdapter.kt** |
| **package** com.example.hellokittyinformation  **import** android.content.Context **import** android.content.Intent **import** android.net.Uri **import** android.os.Bundle **import** android.util.Log **import** android.view.LayoutInflater **import** android.view.View **import** android.view.ViewGroup **import** android.webkit.WebViewClient **import** androidx.core.content.ContextCompat.startActivity **import** androidx.navigation.Navigation **import** androidx.navigation.fragment.NavHostFragment.findNavController **import** androidx.recyclerview.widget.RecyclerView **import** kotlinx.android.synthetic.main.shop\_list.view.\*  *//ShopAdapter 클래스 선언* **class** ShopAdapter(**val context**:Context, **val shop\_DataArray**: ArrayList<ShopData>): RecyclerView.Adapter<ShopAdapter.ItemViewHolder>(){   *//어댑터에서 관리하는 아이템의 개수를 반환* **override fun** getItemCount() = ShopData.values().**size   override fun** onCreateViewHolder(parent: ViewGroup, viewType: Int): ShopAdapter.ItemViewHolder {  **val** adapterView=  LayoutInflater.from(parent.*context*).inflate(R.layout.*shop\_list*, parent, **false**)   **return** ItemViewHolder(adapterView)  }*//end of onCreateViewHolder   //뷰홀더에 데이터를 바인딩* **override fun** onBindViewHolder(holder: ShopAdapter.ItemViewHolder, position: Int) {   *//데이터 바인딩* holder.bindShopData(ShopData.values()[position])    }*//end of onBindViewHolder   //뷰홀더 클래스 선언* **inner class** ItemViewHolder(itemView: View): RecyclerView.ViewHolder(itemView){  *//아이템뷰에 ShopData 데이터 바인딩* **fun** bindShopData(shopData: ShopData){  **itemView**.tv\_item\_shop.*text* = shopData.**shop  itemView**.tv\_item\_info.*text*=shopData.**shopDetail  itemView**.img\_shop.setImageResource(shopData.**imgRes**)  **itemView**.tv\_item\_url.*text*=shopData.**shopUrl** }*//end of bindShopData* }*//end of ItemViewHolder* }*//end of ShopAdapter* |

**6) 타이머**

|  |  |
| --- | --- |
| **화면(UI)** | **설명** |
|  | Chronometer를 통해서 타이머를 구현하였다.  START 버튼 클릭시 타이머가 시작되고 PAUSE 클릭시 잠시 멈추며 RESET 클릭시 다시 0부터 시작한다. |
| **TimerFragment.kt** | |
| **package** com.example.hellokittyinformation.ui  **import** android.os.Bundle **import** android.os.SystemClock **import** android.view.View **import** androidx.fragment.app.Fragment **import** com.example.hellokittyinformation.R **import** android.view.LayoutInflater **import** android.view.ViewGroup **import** kotlinx.android.synthetic.main.fragment\_timer.\*  **class** TimerFragment : Fragment() {   **override fun** onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {   *// Inflate the layout for this fragment* **return** inflater.inflate(R.layout.*fragment\_timer*, container, **false**)  }   **override fun** onViewCreated(view: View, savedInstanceState: Bundle?) {  **super**.onViewCreated(view, savedInstanceState)    **var** stopTime : Long = 0   */\*startBtn을 눌렀을 때 발생하는 이벤트-> 타이머가 작동\*/* startBtn.setOnClickListener**{** chronometer.*base* = SystemClock.elapsedRealtime() + stopTime  chronometer.start();  startBtn.*visibility* = View.*GONE* pauseBtn.*visibility* = View.*VISIBLE* **}** */\*pauseBtn을 눌렀을 때 발생하는 이벤트-> 타이머가 잠시 멈춤\*/* pauseBtn.setOnClickListener**{** stopTime = chronometer.*base* - SystemClock.elapsedRealtime()  chronometer.stop()  startBtn.*visibility* = View.*VISIBLE* pauseBtn.*visibility* = View.*GONE* **}** */\*resetBtn을 눌렀을 때 발생하는 이벤트-> 타이머가 처음부터 재시작\*/* resetBtn.setOnClickListener **{** chronometer.*base* = SystemClock.elapsedRealtime()  stopTime = 0  chronometer.stop();  startBtn.*visibility* = View.*VISIBLE* pauseBtn.*visibility* = View.*GONE* **}** }  } | |

**4. 후기**

**이름: 최희연**

|  |  |
| --- | --- |
|  | **나의 소감**  처음에는 강의 자료에 있는 것을 잘 응용하여 구현하면 될 거라고 쉽게 생각했다. 화면 하나하나를 구현해 나가면서 이게 얼마나 힘든 일인지 알게 되었다. 똑같은 방식으로 작성한 코드도 오류가 생겨 몇 시간을 투자해서 해결한 적도 있었다. 구현해놓고 보니 정말 간단한거 같은데 그 과정은 많은 시간과 노력을 들였다. 나중에 시간적 여유가 생긴다면 더 제대로 된 앱을 개발하고 싶다고 생각했다. |
| **모바일 프로그래밍 강의를 통해 얻은 것**  앱을 만드는 방법에 대해서는 아무것도 몰랐다. 하지만 모바일 프로그래밍 강의를 통해 어떤 원리로 앱을 구현해 내는지 깨달았다. 스마트폰을 사용하면서도 앱을 사용할 때 ‘아 이건 이렇게 구현했겠구나, 이런 방법을 써겠구나, 이런게 어려웠겠구나’ 라고 생각할 수 있게 되었다.  아직은 구현할 수 있는 게 한계가 있지만 더 배워서 제대로 된 앱을 개발해 앱스토어에 실제로 내보고 싶다는 생각도 들었다. 스스로 구현해 봤기 때문에 더 알찼던 강의인 것 같다. | |